verge.io

# THE PROXMOX STORAGE CHALLENGES

# WHY ZFS AND CEPH FORCE IMPOSSIBLE TRADE-OFFS

Prepared by Verge.io

www.verge.io

# The Proxmox Storage Challenges

**Why ZFS and Ceph Force Impossible Trade-offs**

Proxmox storage challenges confront every organization evaluating Proxmox as a VMware alternative—the immediate, consequential decision centers on which storage backend should underpin the environment. Proxmox offers two primary options—ZFS for node-local storage and Ceph for distributed storage—each with fundamentally different architectures, operational models, and trade-offs. Neither option delivers what enterprise environments actually need: unified, efficient, resilient storage that scales without complexity.

This analysis examines the technical realities of both approaches and explains why the Proxmox storage decision exemplifies the broader problem with modular infrastructure architectures.

## Proxmox Storage Challenge #1: ZFS

ZFS provides a copy-on-write filesystem with checksumming to detect silent corruption, inline compression to reduce capacity consumption, and flexible RAID configurations via mirrors and RAIDZ variants. Snapshots are lightweight and instant, enabling quick rollback for VMs and containers on that specific node.

However, ZFS operates as a node-local construct. Each Proxmox node maintains its own ZFS pool using directly attached disks. This architecture creates fundamental limitations for enterprise environments.

### ZFS's Mobility Problem

VMs stored on node-local ZFS cannot simply migrate to another node. The storage doesn't follow the workload. Proxmox addresses this through asynchronous replication using ZFS send/receive. Still, this approach introduces RPO windows—data written since the last replication job can be lost if the source node fails. Even minimum replication intervals of one minute leave data exposed and introduce performance concerns that many production workloads cannot tolerate.

### ZFS's Deduplication Reality

ZFS includes mature inline deduplication, but most Proxmox deployments disable it due to substantial RAM and CPU overhead that competes with VM workloads. Even when enabled, deduplication operates per-node because each node maintains its own independent ZFS pool. The same Windows image deployed across five nodes consumes five times the storage—organizations never achieve the efficiency gains of global deduplication, which eliminates redundant data across an entire cluster.

This limitation compounds in many-to-one DR replication scenarios. When multiple production nodes replicate to a single DR target, each stream arrives independently. The DR site receives identical Windows images, application binaries, and standard data blocks repeatedly from each source—storing them all separately. A ten-node environment could require ten times the logical capacity at the DR target, forcing organizations to either massively overprovision DR storage or implement yet another deduplication layer, adding complexity to an already fragmented architecture.

### ZFS's HA Complexity

High availability with ZFS requires combining multiple independent mechanisms: node-local ZFS for disk resilience, asynchronous inter-node replication for restart capability, and the Proxmox HA manager for failover orchestration. Each layer must be configured, monitored, and maintained separately. When failures occur, recovery depends on replication state, HA configuration, and manual intervention to coordinate across these layers.

## Proxmox Storage Challenge #2: Ceph

### Distributed but Demanding

Ceph takes the opposite approach, providing distributed object storage that all Proxmox nodes access through RBD (Reliable Autonomic Distributed Object Store, RADOS Block Device) volumes that appear as standard disk devices to VMs. This shared storage architecture eliminates the VM mobility problem—workloads migrate freely because storage isn't tied to individual nodes.

Ceph's CRUSH (Controlled Replication Under Scalable Hashing) algorithm determines data placement across configurable failure domains—hosts, racks, or datacenters—without querying a central metadata server. However, **this flexibility comes with overhead**: each client maintains a CRUSH map and performs placement calculations for every I/O operation, consuming CPU cycles that scale with the cluster's complexity. OSD failures or rebalancing will trigger data migrations that compete with production workloads for CPU, network, and disk resources.

CRUSH creates replicated pools with multiple complete copies of each object, or erasure-coded pools using parity chunks for better capacity efficiency. When failures occur, Ceph automatically re-replicates or re-encodes objects to restore redundancy levels.

### Ceph's Complexity Tax

Ceph is a distributed system that demands expertise in distributed systems. Deployment requires understanding placement groups, CRUSH maps, pool configuration, and OSD management. Tuning for performance means balancing replication overhead, network bandwidth, and storage efficiency. Troubleshooting failures requires tracing issues across multiple nodes, network paths, and software layers.

Organizations that adopt Ceph effectively commit to building and maintaining SRE-level (Site Reliability Engineering) expertise for storage infrastructure—the same discipline that Google developed to manage large-scale distributed systems. This means treating storage as a complex software system requiring observability, automation, capacity planning, and incident response practices typically found only in hyperscale operations. The skills necessary differ substantially from traditional storage administration and even from general Linux systems management.

### Ceph's Deduplication Gap

Unlike ZFS, Ceph lacks production-ready deduplication for RBD workloads. The experimental object-level deduplication mechanism in RADOS is not a standard feature for VM storage. Most Proxmox environments running Ceph assume no primary-storage deduplication, accepting the capacity and cost implications.

This gap compounds over time. As VM counts grow and similar workloads proliferate, duplicate data accumulates without reduction. Organizations either accept inflated storage costs or implement separate deduplication at backup layers—adding yet another system to manage.

### Ceph's Redundancy Overhead

Ceph's redundancy models consume substantial raw capacity. Replicated pools with three copies require 3x raw storage for 1x usable capacity. Erasure coding improves efficiency but adds CPU overhead and complicates recovery scenarios. Neither approach matches the storage efficiency achievable with global inline deduplication.

## Proxmox Storage Challenge #3: External AFAs

Given the challenges with both ZFS and Ceph, some organizations consider a third path: connecting Proxmox to external all-flash arrays. This approach returns to traditional siloed infrastructure that virtualized storage was supposed to eliminate.

### Premium Pricing and Proprietary Lock-in

Dedicated arrays bring significant disadvantages. Storage controllers command premium pricing with excessive markups—7X or more—on storage media; organizations pay not just for capacity but also for proprietary hardware that could otherwise be replaced with commodity alternatives. Vendor lock-in at the storage layer limits future flexibility and creates dependency on a single supplier for upgrades, support, and expansion.

### Operational Fragmentation

Separate management interfaces fragment operations, requiring storage administrators to work independently from virtualization teams. Storage professionals comfortable with VMware's vCenter and familiar storage array interfaces face a learning curve with Proxmox's Linux-centric management model, command-line tooling, and different operational paradigms. Organizations must budget for retraining or risk operational disruptions during the transition.

While dedicated arrays can allocate processing power and RAM to features like deduplication without impacting VM workloads, this capability comes at a premium hardware cost that often exceeds the value delivered.

### Scaling Limitations and Infrastructure Complexity

Scaling demands forkliftupgradesrather thanincremental growth; when capacity or performance limits approach, organizations face disruptive replacement cycles rather than simply adding nodes. Organizations also reintroduce single points of failure and network complexity connecting external storage to compute nodes. Storage I/O traverses additional network hops, adding latency and creating bandwidth bottlenecks that integrated architectures avoid.

### Recreating the VMware Problem

Thededicated-arrayapproach essentially recreates the VMware-era architecture—expensive, siloed, and operationally fragmented—while eliminating the cost advantage that attracted organizations to Proxmox in the first place. Organizations seeking to escape VMware's complexity and cost find themselves rebuilding the same infrastructure model with different vendor names on the equipment.

## Proxmox Storage Challenge #4: Data Protection Complications

Regardless of which storage path organizations choose—ZFS, Ceph, or external arrays—Proxmox requires a separate backup solution for comprehensive data protection.

Proxmox Backup Server (PBS) or a third-party alternative remains necessary for VM-level backups with application consistency, granular file-level restores, long-term retention, and off-site replication. External arrays may offer native snapshots and replication, but these storage-level features don't replace the need for proper VM backup capabilities. Organizations choosing the external array path actually add yet another layer to manage—the array's native protection features plus a separate backup solution—further fragmenting operations and increasing complexity.

### The Rehydration Problem

Whenproductionstorageuses deduplication—whether ZFS's inline deduplication, Ceph's experimental deduplication, or an external array's built-in data reduction—data must be rehydrated to full size before being transmitted to PBS, which then performs its own deduplication on the received data. This process creates a dedupe-rehydrate-dedupe cycle that consumes network bandwidth and CPU cycles, extends backup windows, and complicates capacity planning. External arrays compound the problem further by adding network hops between the array, compute nodes, and backup targets—each transfer rehydrates and re-deduplicates data that has already been processed.

Recovery reverses the process, requiring data expansion from PBS, network transfer across storage networks, and reintegration into production storage. Large environments experience recovery times measured in hours or days rather than minutes, and external array architectures introduce latency at each stage of the restore path.

The core of the problem is that these systems lack infrastructure-wide deduplication.

### The Ransomware Exposure

Backup data in PBS resides on external targets that must be secured independently from production infrastructure. Sophisticated ransomware attacks increasingly target backup systems precisely because they represent the last line of defense. Organizations must implement additional security layers, air-gapped storage, and immutability mechanisms—all outside the core Proxmox platform.

## Proxmox Storage Challenge #5: DR Fragmentation

Cross-site resilience with Proxmox requires yet another layer of complexity, with different approaches depending on the storage backend.

### ZFS DR Approach

ZFS-backed environments typically combine asynchronous replication via ZFS send/receive with backup-based DR through PBS. RPO depends on replication intervals, while RTO depends on failover automation and runbook execution. Multi-site resilience requires maintaining parallel infrastructure, replication jobs, and coordination procedures.

### Ceph DR Approach

Ceph offers two DR patterns, each with distinct trade-offs.

RBD mirroring between independent Ceph clusters provides asynchronous block-level replication. Journal-based mirroring achieves near-real-time RPO at the cost of higher I/O overhead, while snapshot-based mirroring ties RPO to the snapshot frequency. Failover requires promoting the mirrored images on the DR cluster and starting the VMs in the DR Proxmox environment.

Stretch clusters span a single Ceph cluster across sites with synchronous inter-site replication, achieving RPO of zero for site failures. However, this approach demands low-latency, high-bandwidth links between sites and careful failure-domain design. Complexity increases substantially, and network requirements often prove prohibitive.

### External Array DR Approach

External arrays introduce their own DR mechanisms that operate independently from Proxmox. Array-based replication—whether synchronous or asynchronous—replicates storage volumes between sites but requires matching arrays at both locations, doubling hardware investment and vendor lock-in.

Organizations must coordinate array-level replication with Proxmox VM configurations to ensure replicated storage volumes align with VM definitions at the DR site. Failover involves promoting replicated volumes on the secondary array, reconfiguring Proxmox to recognize the new storage paths, and starting VMs—a multi-step process spanning separate management interfaces. Many organizations still layer PBS on top of array replication for application-consistent recovery points, creating yet another coordination requirement between storage replication schedules and backup windows.

## The Operational Reality

Organizations adopting Proxmox face an ongoing operational burden that extends far beyond initial deployment, regardless of which storage path they choose.

**Multiple management interfaces** require attention: Proxmox for compute, ZFS or Ceph tooling for storage (or the external array's proprietary management console), PBS for backup, and potentially additional tools for networking and security. External arrays add yet another interface with its own authentication, alerting, and reporting systems that don't integrate natively with Proxmox workflows.

**Disparate update cycles** create maintenance windows and compatibility concerns. ZFS, Ceph, Proxmox, and PBS each release updates independently, and organizations must validate combinations before production deployment. External arrays introduce firmware updates, microcode patches, and feature releases on the vendor's schedule—often requiring maintenance windows that conflict with hypervisor updates and occasionally breaking compatibility with existing configurations.

**Fragmented troubleshooting** complicates incident response. When VMs experience storage performance issues, the root cause might lie in ZFS pool configuration, Ceph placement group health, network congestion, PBS backup activity, or—with external arrays—controller bottlenecks, fabric congestion, or array-side deduplication overhead. Diagnosis requires expertise across multiple domains and coordination between teams who may not share standard tooling or terminology.

**Skill requirements multiply** as each layer demands distinct expertise. Finding staff who understand ZFS internals, Ceph distributed systems concepts, Proxmox hypervisor management, and backup infrastructure represents a significant hiring challenge. External arrays demand additional vendor-specific certifications and training, further fragmenting team expertise and increasing dependency on specialized—and expensive—storage administrators.

## What Organizations Actually Need

The Proxmox storage challenges reflect a broader architectural problem. Modular infrastructure forces trade-offs that unified platforms eliminate.

Organizations migrating from VMware need storage that provides global inline deduplication without the RAM overhead that makes ZFS deduplication impractical. They need distributed storage without the operational complexity that makes Ceph demanding. They need integrated data protection without the rehydration cycles that slow backup and recovery. They need resilience without the multi-layer coordination that characterizes Proxmox HA configurations.

These requirements point toward Infrastructure Operating Systems that unify compute, storage, networking, and data protection within a single platform. Rather than choosing between ZFS's simplicity and Ceph's distribution, organizations can deploy unified storage that delivers both characteristics natively.

Global deduplication across all workloads and all nodes reduces storage consumption by 60-80% compared to non-deduplicated approaches—without the per-node RAM overhead that limits ZFS dedupe adoption. Shared metadata architecture eliminates rehydration during backup and recovery, enabling snapshot operations that complete in seconds regardless of data volume. Linear scalability with flexible node roles—balanced, storage-only, or compute-only—provides growth options that neither ZFS nor Ceph alone can match.

## How VergeOS Eliminates the Proxmox Storage Challenges

Proxmox introduces complexity because its storage layers were never designed to operate as a unified system. ZFS protects a single node with strong integrity features, but fragments storage across the cluster. Ceph distributes data across nodes but demands a specialized engineering discipline. External arrays centralize storage but recreate the silos that many organizations want to leave behind. These problems share a root cause: each storage model requires a separate management domain, protection framework, and operational philosophy. VergeOS avoids this fragmentation by placing storage within the Infrastructure Operating System itself.

### Unified Storage Instead of Fragmented Backends

VergeFS provides a global storage model spanning every storage contributing node in a cluster. There are no pools tied to individual hosts, nor are there storage networks to build or tune.

Storage becomes part of the operating environment rather than an external subsystem. This design eliminates the mobility limitations that ZFS introduces—a VM stored on VergeFS can move freely because every node sees the same metadata, block references, and deduplication catalog. Live migration becomes a function of the operating environment rather than a scripted replication process that trails the workload.

Organizations can deploy balanced nodes that contribute both compute and storage, or designate storage-only and compute-only nodes to match workload requirements—flexibility neither ZFS nor Ceph provides natively. Storage scales linearly as nodes are added, with the platform automatically distributing data across available resources without CRUSH maps to configure, placement groups to tune, or OSD management overhead.

## Global Deduplication Without Overhead

The deduplication architecture in VergeOS addresses the per-node constraints found in ZFS and the lack of functionality in Ceph. Global inline deduplication covers every block across every workload. The same Windows image used across twenty VMs consumes the space of a single instance. Application binaries, operating system files, and standard datasets are collapsed into shared references. This reduction applies across production sites, DR targets, and archive locations—lowering raw storage requirements throughout the lifecycle without increasing RAM pressure on compute nodes.

In many-to-one DR scenarios, VergeOS transmits unique deduplicated blocks between sites. Ten production nodes replicating to a single DR target send only actual unique data, not ten redundant copies of standard blocks. Organizations right-size DR storage based on actual unique capacity rather than massively overprovisioning for replicated redundancy.

## Failure Tolerance Designed for Real Events

Failure tolerance in VergeOS reflects an architectural decision to sustain operations during events that exceed predictable limits. ZFS handles drive failures within a vdev. Ceph rebuilds OSDs across the cluster. Both approaches attempt to restore redundancy after an event. VergeOS maintains operation during the event.

ioGuardian stores an independent third copy of data outside the production instance. If enough drives or nodes fail to exceed local protection, VergeOS retrieves only the missing blocks from ioGuardian. VM operations continue during these events without waiting for reconstruction. Large drive sizes and multi-node failures stop being risk multipliers because the architecture avoids lengthy rebuilds that plague traditional RAID and erasure coding approaches.

## Integrated Data Protection Without Rehydration

VergeOS addresses data protection through integrated snapshots, ioReplicate, and ioFortify—eliminating the dedupe-rehydrate-dedupe cycle that plagues Proxmox backup operations. Because snapshots leverage the same global deduplication architecture as production storage, creating a snapshot is essentially a metadata operation that completes in seconds regardless of data volume. There's no data movement, no rehydration, no re-deduplication at a separate backup target.

Snapshots become independent, immutable clones that can be mounted instantly, rolled back in seconds, or repurposed for testing and development. Recovery from ransomware involves advancing metadata to a known-good point—an operation that completes in seconds even for 100TB or 100PB environments. Organizations gain protection capabilities that require PBS, plus additional security layers, in Proxmox, delivered natively without bolt-on components.

## Disaster Recovery Built Into the Platform

Disaster recovery follows a single architectural pattern in VergeOS. Administrators create a Virtual Data Center at the DR location. ioReplicate sends deduplicated block changes to that VDC. VM configurations, networking rules, storage references, and protection policies all remain consistent because they operate within the same Infrastructure Operating System.

Recovery does not require aligning Ceph mirroring with Proxmox schedules or mapping array-based replication to VM definitions. DR becomes an extension of the platform rather than a separate engineering project spanning multiple vendors and management interfaces.

## No External Arrays Required

VergeOS eliminates the need for external all-flash arrays. Storage integrates directly into the Infrastructure Operating System, removing premium controller costs, proprietary media markup, separate management interfaces, and forklift upgrade cycles. Organizations leverage commodity servers and standard storage media while achieving better efficiency through global deduplication than dedicated arrays can deliver at premium prices.

Like external arrays, VergeOS can scale compute and storage independently—but without the architectural penalties. Organizations can deploy storage-only nodes for capacity-intensive workloads and compute-only nodes for processing-heavy applications, scaling each resource dimension based on actual demand rather than fixed ratios. This flexibility matches what dedicated arrays offer while eliminating vendor lock-in, premium pricing, and operational fragmentation introduced by external storage.

## Single Management Interface, Single Vendor

Operational simplicity emerges from the unified approach. VergeOS maintains one update cycle for compute, storage, networking, and protection. Troubleshooting becomes faster because every component exists in the same codebase. Engineers no longer need deep ZFS knowledge, Ceph distributed systems training, or vendor-specific array certifications.

Where Proxmox demands expertise across multiple domains with different tooling and terminology, VergeOS consolidates operations into a single platform with unified management.

Enterprise-grade 24x7 support provides single-vendor accountability for the entire infrastructure stack. When issues arise, organizations contact one support team with visibility across all infrastructure functions—not separate vendors pointing fingers between hypervisor, storage, backup, and networking layers.

This unified model eliminates the structural weaknesses that ZFS, Ceph, and external arrays introduce into Proxmox environments. VergeOS delivers the mobility, protection, resilience, and data efficiency that organizations expect from a production platform without asking administrators to assemble these capabilities from separate tools.

## Private AI Ready

VergeOS includes VergeIQ, an integrated Private AI platform that positions organizations for emerging AI requirements. While Proxmox environments would require assembling separate Kubernetes orchestration, GPU passthrough configuration, and external storage systems to support AI workloads, VergeIQ delivers conversational AI interfaces to proprietary data within minutes of installation. GPUs across the cluster are pooled without specialized licensing, and global deduplication extends to AI datasets and model embeddings. Organizations gain AI capabilities as a native platform feature rather than another integration project.

Storage Platform Comparison

| Capability | ZFS | Ceph | External AFA | VergeOS |
|---|---|---|---|---|
| Deduplication | Per-node only; high RAM overhead | Experimental; not production-ready | Array-level; premium cost | Global inline; no overhead |
| VM Mobility | Requires async replication | Native shared storage | Native shared storage | Native shared storage |
| Operational Complexity | Moderate | High (SRE-level) | High (separate management) | Low (unified platform) |
| Scaling Model | Per-node pools | Distributed; CRUSH overhead | Forklift upgrades | Linear; flexible node roles |
| Data Protection | Requires PBS | Requires PBS | Requires PBS + array snapshots | Integrated snapshots |
| DR Approach | ZFS send/receive + PBS | RBD mirroring or stretch clusters | Array replication + PBS | ioReplicate (deduplicated) |
| Rehydration Required | Yes | Yes | Yes | No |
| Ransomware Recovery | Hours/days | Hours/days | Hours/days | Seconds |
| Management Interfaces | Multiple | Multiple | Multiple | Single |
| Vendor Accountability | Community + multiple vendors | Community + multiple vendors | Multiple vendors | Single vendor |
| Hardware Requirements | Commodity + high RAM | Dedicated nodes | Premium arrays | Commodity servers |
| TCO | Hidden complexity costs | High operational costs | Premium hardware + ops | Lowest total cost |
| AI/ML Ready | DIY assembly required | DIY assembly required | DIY assembly required | Native VergeIQ |

## Conclusion

Proxmox presents organizations with a storage choice that has no good answer. ZFS delivers strong local storage but sacrifices mobility, cluster-wide resilience, and practical deduplication. Ceph provides distributed storage but demands SRE-level expertise that most organizations lack, and operates without production deduplication. External all-flash arrays restore traditional siloed architecture with premium costs, vendor lock-in, additional management interfaces, and the same operational fragmentation that drove organizations away from VMware in the first place.

All three options require Proxmox Backup Server or a third-party alternative as a separate layer, introducing rehydration inefficiencies and external targets that complicate recovery and create security exposure. All three options demand multi-layer coordination for high availability and disaster recovery, with external arrays adding yet another vendor's replication mechanisms to coordinate across sites.

The fundamental problem is architectural. Proxmox assembles infrastructure from independent modular components, each optimized for different objectives and managed through separate interfaces. The integration burden falls on customers, who must develop expertise across multiple domains, retrain staff on unfamiliar paradigms, and accept trade-offs that unified platforms avoid entirely.

Organizations evaluating Proxmox as a VMware alternative should understand these storage realities before committing. The zero licensing cost that makes Proxmox attractive conceals operational complexity, training requirements, and capability limitations that become apparent only after deployment. For production enterprise workloads, the Proxmox storage challenges alone often justify evaluating Infrastructure Operating Systems that eliminate these impossible trade-offs by design.