Experimental Evaluation of Open-Source and Proprietary HCI Platforms for Artificial Intelligence Workloads

Rodolfo Wanderley*, Caio Souza*‡, Israel Duarte*, Michaely Silva*, Andson Balieiro*, Kelvin Dias*
*Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Recife, Brazil.
‡ Universidade de Pernambuco (UPE), Surubim, Brazil.
{rjpw, cbbs, ibsd, amb4, kld}@cin.ufpe.br, caiobruno.souza@upe.br, michaely.raiany@ufpe.br

***Abstract*—Deploying Hyperconverged Infrastructure (HCI) for artificial intelligence (AI) workloads requires balancing low-latency performance against Total Cost of Ownership (TCO). Choosing between proprietary solutions like Nutanix AOS and open-source stacks like Proxmox VE with Ceph Storage hinges on whether performance gains justify licensing costs. This paper presents an empirical evaluation comparing both architectures across AI workloads, including dataset loading, checkpointing, Feature Stores, and cluster stress, linking performance directly to their storage cache designs. Nutanix uses automated RAM/SSD Cache Tiering with Data Locality, whereas Ceph BlueStore isolates metadata and Write-Ahead Logs on SSDs. Results confirm Nutanix's superiority in small-block random I/O and high-concurrency tasks, maintaining tight latency bounds to prevent GPU starvation and request timeouts. Conversely, while Proxmox/Ceph exhibits replication bottlenecks under random stress, it operates efficiently during large-block sequential writes. Consequently, Nutanix is optimal for latency-critical GPU training and inference pipelines, whereas Proxmox VE provides a cost-effective platform for secondary AI stages, enabling capital reallocation toward hardware accelerators.**

***Index Terms*—Hyperconverged Infrastructure, Nutanix AOS, Proxmox VE, Ceph Storage, AI Workloads, Performance Evaluation**

I. Introduction

Educational and research institutions face severe budgetary constraints when procuring the on-premises infrastructure required to host both core institutional applications and demanding research environments, such as those dedicated to the development and training of Artificial Intelligence (AI) models, where costs escalate rapidly alongside growing computational demand [1]. In scientific research, experimental outcomes are highly sensitive to data processing capacity and throughput, underscoring the need for robust computational support to ensure adequate performance and the ultimate reliability of research findings [2].

To mitigate these constraints, research institutions often deploy shared data centers to support concurrent workloads [1]. While traditional three-tier architectures separating compute hosts, high-speed networking, and dedicated storage arrays offer strong performance, their high capital expenditure often limits adoption. Alternatively, Hyperconverged Infrastructure (HCI) eliminates dedicated SANs and storage appliances by consolidating compute, networking, and storage into software-defined x86 servers [3]. Prominent implementations range from commercial platforms like Nutanix to open-source stacks combining Proxmox VE and Ceph.

Open-source solutions offer extensive flexibility and customizability for specific operational environments [4]. Conversely, commercial alternatives typically prioritize high performance alongside native management and automated fault-detection capabilities [5]. Despite these general distinctions, it remains unclear whether proprietary licensing fees translate into justifiable performance gains for AI model training within shared environments. Consequently, selecting an optimal HCI platform requires a comprehensive evaluation that weighs operational capabilities well beyond upfront acquisition costs.

Existing virtualization performance studies exhibit notable limitations. Many overlook deployment and acquisition costs, relying instead on conceptual reviews, subjective metrics [6], or basic workload benchmarks comparing proprietary and open-source platforms [7]–[9]. Others adopt narrow scopes, evaluating isolated CPU metrics [10] or file system performance [7], [8] without correlating HCI throughput with implementation costs. Furthermore, several analyses depend on mathematical formalisms or tightly constrained experimental setups, such as performance under induced interference [6], to model system behavior [11], [12], ultimately limiting the generalizability of their findings to real-world production environments.

To address these gaps, this paper presents a comprehensive comparative study of HCI platforms deployed on enterprise-grade hardware, evaluating whether commercial solutions deliver tangible storage performance advantages or if an open-source alternative can adequately support the intensive I/O demands of AI model training workloads across virtual machines. The analysis evaluates Input/Output Operations Per Second (IOPS), throughput, and data access latency to provide a thorough performance assessment of each platform. Results show that while Nutanix significantly outperforms Proxmox VE in high-concurrency and random I/O workloads, Proxmox VE with Ceph remains a highly viable, cost-effective alternative for sequential, large-block operational tasks such as model checkpointing and file sharing.

The remainder of this paper is organized as follows: Section II reviews related work; Section III details the experimental setup and testing protocol; Section IV evaluates the obtained

results; and Section V concludes the study and outlines potential future research directions.

II. Related Work

Virtualization environments for application hosting have been extensively studied due to the ongoing need for infrastructure optimization. However, current literature exhibits a notable methodological and practical disconnect. As highlighted by [6], many evaluations rely strictly on conceptual reviews and subjective scoring, failing to capture the architectural complexity of modern hypervisors. Other studies focus primarily on compute performance [10], [13]. While [10] conducts an empirical comparison of absolute execution times to categorize Hyperconverged Infrastructures (HCI), [13] evaluates performance and architectural impact using analytical metrics of total processing time.

Targeting file system dynamics, [7] and [8] propose mathematical evaluation models. Using the Filebench tool across four workload profiles, [7] compares empirical metrics against model-predicted execution times. Conversely, [8] and [11] leverage execution time and throughput to assess processing capacity. Expanding upon this, [12] evaluates hypervisors across Input/Output (I/O) rates, execution time, and network bandwidth, while also accounting for platform concurrency and stability. Despite these contributions, existing literature lacks a robust, multifaceted analysis linking HCI performance directly to implementation costs. For instance, studies such as [10] rely almost exclusively on CPU benchmarks, neglecting potential bottlenecks in other physical or virtual components.

Furthermore, while comparing proprietary and open-source HCIs is highly relevant, studies like [7] and [8] restrict their scope to direct workload-benchmark correlations, whereas [9] confines its parameters to file system metrics through a purely conceptual framework. Finally, many of these efforts depend on abstract mathematical formalisms or highly constrained experimental setups, such as evaluation under induced interference [6] or strict modeling of isolated configurations [11], [12], limiting the generalizability of their findings to real-world production environments.

While analytical approaches in synthetic scenarios overlook the relationship between I/O performance and operational efficiency in enterprise clusters, this study evaluates the direct impact of hyperconverged storage on Artificial Intelligence pipelines. Specifically, our contribution empirically contrasts an open-source stack (Proxmox VE + Ceph) with a commercial platform (Nutanix AOS) on identical enterprise hardware, correlating storage access patterns (random dataset reading, sequential checkpoint writing, and concurrent Feature Store access) with the internal caching and data locality mechanisms of each platform.

III. Evaluation Methodology

The methodology adopted for the comparative performance evaluation of hyperconverged infrastructure (HCI) solutions is grounded in the principles of the HCIBench benchmark [14], employing the Flexible I/O Tester (FIO) tool for synthetic workload generation, distribution, and measurement. The experimental workflow was structured as an automated, centralized client-server architecture, where a controller instance orchestrates concurrent task execution across operational instances distributed throughout the cluster. The evaluation encompasses multiple operational scenarios representative of enterprise and research environments, executed in controlled, sequential phases. This approach ensures experimental reproducibility, steady-state measurement stability, and bypasses artificial memory caching mechanisms that could otherwise mask true storage performance.

A. Experimental Architecture and Orchestration

To implement the HCIBench-based experimental protocol [14], 11 Rocky Linux virtual machines (VMs) were deployed on each evaluated HCI platform: 10 client (worker) instances dedicated to workload processing and 1 controller (server) instance responsible for central orchestration. Leveraging FIO's native client-server mode eliminates the need for external execution scripts and enables automated, aggregated performance data collection.The controller VM manages testing by issuing concurrent requests to the background FIO daemon running on the client instances. Synthetic workloads are uniformly distributed across the cluster and executed in four distinct phases. Upon completing each experiment, FIO natively consolidates results from all nodes into a unified global report, yielding precise metrics for total IOPS, aggregated throughput, and latency percentiles.

To reflect realistic operational stress conditions, the configuration file (`job.fio`) enforces strict directives outlined in Table I. An asynchronous I/O engine (`libaio`) is used to issue multiple concurrent requests, while direct I/O (`direct=1`) bypasses system RAM to prevent artificial cache acceleration. Disabling predictable access patterns (`randrepeat=0` and `norandommap=1`) prevents file system pre-fetching optimizations. To guarantee steady-state evaluation, each test runs for 3600,s (1 hour), with the initial 60,s ramp-up time discarded to eliminate transient initialization noise. Transitions between scenarios are governed by the `stonewall` parameter, which serves as a synchronization barrier, ensuring a new phase begins only after the complete finalization of the preceding phase across all concurrent instances.

TABLE I
Global FIO Experimental Settings

Parameter	Setting
I/O Engine	`libaio`
Direct I/O	`direct=1`
Runtime per Phase	3600 s (1h)
Ramp Time	60 s
Random Repeat	`randrepeat=0`
Random Mapping	`norandommap=1`
Phase Separation	`stonewall`
Total Volume per Phase	400 GB total (10 VMs)

The physical and virtualized evaluation environment is summarized in Table II. The underlying infrastructure comprises six Lenovo ThinkAgile HX5521 servers partitioned into two distinct, three-node HCI clusters: a Nutanix-based deployment

and a Proxmox VE deployment backed by Ceph Storage. Across both setups, the local storage devices of the physical nodes are aggregated into a distributed storage pool, dynamically sharing capacity, IOPS, and compute resources. High-speed network connectivity and inter-node communication are sustained by Extreme Networks X670V switches configured with dual 10,GbE links per host, as illustrated in Fig. 1.

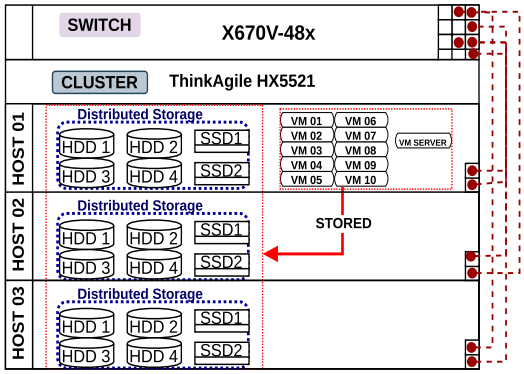


Fig. 1. Architecture of both clusters HCI testbeds.

TABLE II
INFRASTRUCTURE, HCI, AND VIRTUAL MACHINE CONFIGURATIONS

Hosts (3x Lenovo)	
Model	ThinkAgile HX5521 Node
Processor	2 x Intel(R) Xeon(R) 4216
Cores	16 Cores per processor
Memory	12 x 32 GB DDR4 (384 GB)
Storage Disks 1	2x SSD SATA 1.92TB
Storage Disks 2	4x HDD SATA 6TB
Switches (Xtreme)	
Switch Model	X670V-48x-4q
Switch Base Ports	48x SFP+ (1/10Gb) + 4x QSFP+ (10/40Gb)
HCI (Hyperconverged Infrastructure)	
Nutanix PC Version	pc.2022.6.0.9
Nutanix NCC Version	4.6.0
Nutanix LCM Version	2.7.1
Proxmox VE	Version 8.3.0
Ceph Storage	Version 19.2.0
VMs (Virtual Machines)	
Operating System	Rocky Linux 10.2
vCPU	4 vCPUs
RAM	8 GB
Storage Allocation	3 TiB Disk

IV. RESULT ANALYSIS

To evaluate hyperconverged infrastructure (HCI) performance under high-demand enterprise and scientific workloads, four operational scenarios were designed, generating a cumulative total of approximately 516.6 million I/O operations over 8 hours of stress across the clusters. The corresponding FIO execution parameters are summarized in Table III. The first scenario, AI Dataset Loading, evaluates throughput during random read operations across medium-to-large blocks to quantify processing capacity during concurrent unstructured data access by machine learning algorithms. Next, the AI Checkpoint Saving phase measures peak write throughput and storage-tier saturation limits under a massive, continuous sequential stream, mirroring the periodic persistence of model states. The File Server profile simulates mixed sequential operations under moderate concurrency to assess latency and multi-user responsiveness in shared directories, while replicating direct training data transfers to virtual machines. Finally, the Database scenario evaluates raw IOPS processing and transactional latency through small-block random requests that stress storage controllers and caching tiers; this profile simulates structured-data AI pipelines, such as tabular datasets, relational databases, or feature stores.

TABLE III
FIO BENCHMARK PROFILE AND EXECUTION CONFIGURATIONS

AI Dataset Loading	
Access Pattern	Random (`randrw`)
Read / Write Ratio	90% / 10%
Block Size Range	32 KB – 256 KB
Jobs & I/O Depth	4 jobs / Depth 4
Data Volume	10 GB/job (40 GB/VM)
AI Checkpoint Saving	
Phase Barrier	`stonewall`
Access Pattern	Sequential Write (`write`)
Read / Write Ratio	0% / 100%
Block Size	1 MB
Jobs & I/O Depth	2 jobs / Depth 4
Data Volume	20 GB/job (40 GB/VM)
File Server	
Phase Barrier	`stonewall`
Access Pattern	Sequential (`rw`)
Read / Write Ratio	80% / 20%
Block Size Range	4 KB – 64 KB
Jobs & I/O Depth	2 jobs / Depth 2
Data Volume	20 GB/job (40 GB/VM)
Database (OLTP)	
Phase Barrier	`stonewall`
Access Pattern	Random (`randrw`)
Read / Write Ratio	70% / 30%
Block Size	8 KB
Jobs & I/O Depth	2 jobs / Depth 4
Data Volume	20 GB/job (40 GB/VM)

In evaluating the dataset loading task, with average per-virtual-machine metrics shown in Fig. 2, Nutanix demonstrated substantial superiority over Proxmox in handling medium-to-large block random I/O. Nutanix achieved approximately 800% higher read throughput ($9\times$) and 689% higher IOPS ($7.9\times$). The most critical disparity emerged in $p99$ write latency, where Proxmox exhibited response times approximately 21 times higher (3895.67,ms versus 185.90,ms for Nutanix). This behavior indicates that under concurrent, read-intensive unstructured data access, Proxmox's storage layer experienced severe queue saturation, leading to critical synchronization bottlenecks.

In the context of deep learning pipelines, this performance gap directly impacts hardware accelerator utilization. Proxmox's extreme latency induces I/O Wait, leaving Graphics Processing Units (GPUs) idle while awaiting data batches from storage. This bottleneck significantly extends training epoch durations and inflates overall computational costs. Conversely, Nutanix maintained the stability necessary to sustain a

continuous data stream to training virtual machines, ensuring consistent GPU feeding without operational stalls.

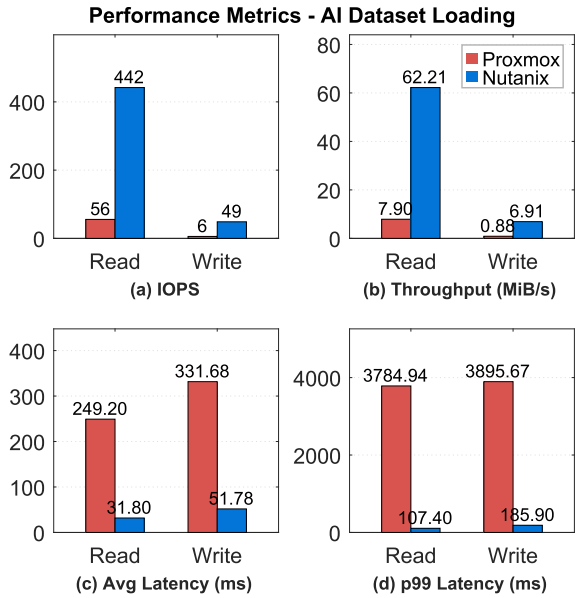


Fig. 2. Performance metrics for the AI Dataset Loading workload.

In the checkpoint saving scenario, evaluated via average per-virtual-machine metrics shown in Fig. 3 and centered on high-volume, 1MB sequential writes, the performance gap between platforms narrowed, though Nutanix retained its lead. Nutanix achieved 113% higher write throughput and IOPS, more than double that of Proxmox. Regarding response times during write bursts, Nutanix reduced average latency by 52.7% compared to Proxmox (173.80,ms versus 367.53,ms). This scenario represented Proxmox's strongest relative performance, as Ceph handles continuous, large-block sequential streams considerably more efficiently than small random workloads, demonstrating its ability to absorb massive data bursts despite operating at half the speed of its enterprise competitor. In practical AI deployments, periodic checkpointing is essential to persist model weights during long-running training routines. Nutanix's lower latency and higher throughput effectively halve virtual machine pause times, minimizing the duration training is halted for network state persistence. Nevertheless, Proxmox/Ceph's performance in this benchmark confirms its viability as a secondary repository for checkpoints and model versioning in small-to-medium training workflows, where slightly extended sequential write windows do not critically impede experimental progress.

The file server simulation, which evaluates average per-instance behavior as illustrated in Fig. 4, incorporates mixed concurrency and routine operational access patterns, maintaining Nutanix's advantage at a 3×1 margin. Nutanix recorded 205% higher read and write IOPS ($3.05\times$) and 204% higher read throughput compared to Proxmox. Latency analysis (both average and $p99$) confirms that Nutanix's architecture provides more responsive and predictable performance under

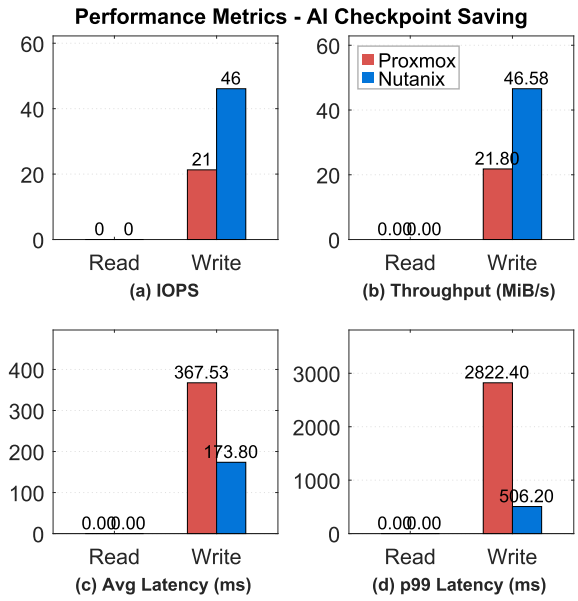


Fig. 3. Performance metrics for the AI Checkpoint Saving workload.

concurrent multi-user access. Nevertheless, Proxmox sustained stable operation without severe degradation, demonstrating its capability to handle conventional shared storage demands.

In a data science environment, the file server facilitates sharing scripts, virtual environments, orchestrator logs (e.g., MLflow), and shared data preparation directories. Nutanix's performance ensures seamless, low-latency access for machine learning engineering teams working concurrently on the same cluster. Conversely, the stability demonstrated by Proxmox in this scenario confirms its efficacy for hosting infrastructure support tools, raw data repositories, and academic collaboration environments, satisfactorily serving concurrent access without incurring the substantial financial burden of proprietary licensing.

The transactional database workload, with average per-virtual-machine metrics consolidated in Fig. 5, exhibited the most significant performance divergence between the evaluated platforms. Nutanix outperformed Proxmox by achieving 1000% higher read and write IOPS ($11\times$) alongside 133% higher overall throughput. The most critical drawback for Proxmox emerged in $p99$ latency, which reached 544.70 ms, a response time 22.9 times higher (2294% greater) than Nutanix's 23.75,ms. While Nutanix processed thousands of small 8,KB random requests with high stability across its caching layers, the Proxmox/Ceph architecture struggled under high queue depth, yielding latency spikes that would compromise time-sensitive transactional applications. This scenario directly mirrors the demands of modern AI architectures relying on structured data, such as relational or vector Feature Stores and retrieval pipelines for Retrieval-Augmented Generation (RAG). Proxmox's $p99$ latency exceeding half a second renders it unsuitable for real-time AI inference systems, where models must query contextual variables from databases and re-

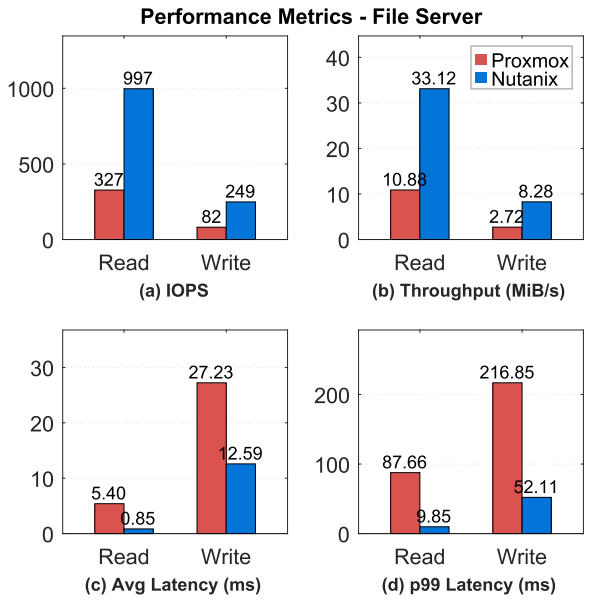


Fig. 4. Performance metrics for the File Server workload.

spond within milliseconds. Spikes of this magnitude would inevitably trigger application-level request timeouts. Conversely, Nutanix's consistently low latency validates its architecture for hosting high-concurrency Feature Stores and mission-critical inference microservices in production environments.

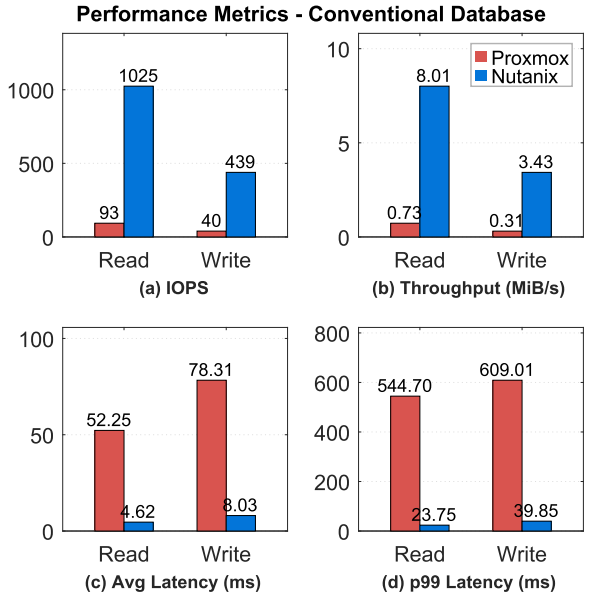


Fig. 5. Performance metrics for the Conventional Database workload.

Beyond individual virtual machine evaluations, observing the aggregated cluster-wide performance, as summarized in Fig. 6, reveals the full impact of distributed scaling on the physical infrastructure. In terms of total IOPS, Nutanix achieved an exceptionally high rate of $24,637$ read IOPS and $7,824$ write IOPS, vastly outperforming Proxmox, which registered only 476 read IOPS ($51.7\times$ lower, representing a 5075% advantage for Nutanix) and $1,483$ write IOPS ($5.27\times$ lower). This disparity directly translates to total aggregated throughput: Nutanix sustained 1033.39,MiB/s in read and 652.80,MiB/s in write operations, whereas Proxmox delivered 195.13,MiB/s in read ($5.29\times$ lower) and 257.13,MiB/s in write ($2.54\times$ lower).

The systemic bottleneck of Proxmox/Ceph under maximum load becomes even more pronounced when examining global latency metrics in Fig. 6. While Nutanix maintained global average latencies of 7.97,ms for read and 21.97,ms for write operations, Proxmox recorded 43.06,ms ($5.4\times$ higher) and 101.52,ms ($4.6\times$ higher), respectively. However, Proxmox's performance degradation manifests most acutely in tail percentiles ($p99$): Proxmox reached a $p99$ read latency of 3784.94 ms 35.2 times higher than Nutanix's 107.40 ms, and a $p99$ write latency of 3895.67,ms, compared to 506.20,ms for Nutanix ($7.7\times$ higher). This macro-level perspective demonstrates that under severe, concurrent stress across all instances, the Ceph architecture suffers from extreme network contention and inter-node journal synchronization bottlenecks, leading to response stalls approaching four seconds. Conversely, Nutanix's Data Locality architecture effectively absorbs traffic linearly while keeping latency variance tightly bounded.

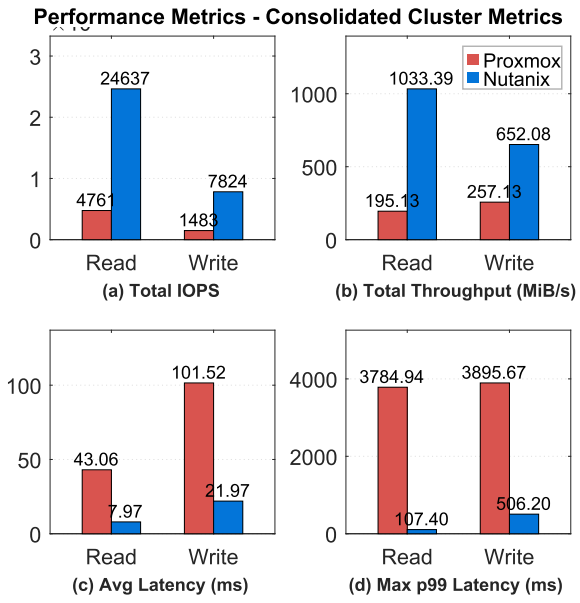


Fig. 6. Comparison of consolidated cluster performance metrics.

The consolidated empirical findings confirm Nutanix's architectural superiority across all evaluated benchmarking scenarios. Operating on identical hybrid storage topologies (SSD acceleration over rotational HDD capacity), Nutanix AOS leveraged Data Locality and intelligent caching to effectively bypass inter-node fabric latency under high-concurrency workloads. Conversely, despite sharing the same SSD acceleration layer, Proxmox VE with Ceph Storage exposed structural

limitations during small-block, random I/O operations. Under severe concurrent stress, Ceph's distributed write-replication protocol and journal synchronization caused heavy queue saturation and pronounced tail latency spikes.

These underlying storage dynamics directly govern hyperconverged platform feasibility within modern AI pipelines. In read-heavy dataset loading, Proxmox's elevated tail latency induces significant I/O Wait states on training VMs, forcing GPUs to idle, prolonging training epochs, and inflating compute costs. In transactional workloads modeling relational or vector Feature Stores, Proxmox's latency instability risks triggering request timeouts, rendering it ill-suited for low-latency Retrieval-Augmented Generation (RAG) inference systems. Conversely, Nutanix maintains the throughput stability and tight latency bounds required to keep accelerator pipelines fully saturated and ensure seamless real-time model serving.

While Proxmox and Ceph excel at large-block sequential writes (e.g., model checkpointing) and mixed-concurrency file serving, aggregated cluster stress creates network contention and journal synchronization bottlenecks. Conversely, Nutanix AOS maintained bounded latency in the tested three-node cluster via Data Locality, though scaling evaluations on larger topologies remain necessary. Financially, adopting open-source Proxmox VE with Ceph eliminates Nutanix licensing fees, but TCO depends on balancing OPEX, tuning overhead, and I/O efficiency. Thus, a multi-tiered infrastructure is recommended: Nutanix AOS for latency-critical workloads (GPU training, Feature Stores, production inference), and Proxmox VE with Ceph for secondary AI lifecycle stages (data lakes, batch preparation, checkpoints, log tracking, and academic prototyping).

V. Conclusions and Future Directions

This paper presented an experimental evaluation comparing open-source Proxmox VE with Ceph Storage against commercial Nutanix AOS to determine whether proprietary performance gains justify licensing costs in AI environments. By benchmarking throughput, IOPS, and tail latency, this study directly links workload behavior to underlying storage cache architectures: Nutanix's automated Cache Tiering, which dynamically migrates hot data between flash and rotational drives, versus Ceph BlueStore, which isolates metadata and Write-Ahead Logs (DB/WAL) on SSDs while committing primary data to HDDs. Results confirm Nutanix's technical superiority in small-block random I/O and high-concurrency tasks, where its Data Locality mechanism bypasses network latency to prevent GPU starvation and Feature Store timeouts. Conversely, while Proxmox/Ceph exhibited replication bottlenecks and latency spikes under random write stress, it performed efficiently in large-block sequential operations like model checkpointing. Consequently, Nutanix is best suited for primary, latency-critical GPU training and inference pipelines, whereas Proxmox VE provides a cost-effective solution for secondary AI lifecycle stages, allowing organizations to reallocate licensing budgets toward hardware accelerators.

Future research directions should evaluate newer Ceph releases incorporating Crimson OSD to assess its ability to replace BlueStore, eliminate CPU bottlenecks, and mitigate latency spikes under severe stress. Additionally, expanding this benchmark framework to include other enterprise hyperconverged platforms, such as VMware vSAN and ESXi, will offer datacenters a broader multi-platform performance matrix for next-generation infrastructure planning.

References

[1] A. Ganzenmüller, P. Rodrigues, V. Kushnarenko, M. Puskaric, and T. Bönisch, "The challenges of setting up sustainable research data infrastructure," in *Proceedings of the 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2025, pp. 1–5. [Online]. Available: https://doi.org

[2] A. R. Group, "High-performance computing for scaling large-scale language and data models in enterprise applications," *Journal of Advanced Scientific Research and Conference (ASRC)*, vol. 1, no. 1, pp. 60–72, 2023. [Online]. Available: https://global.asrcconference.com/index.php/asrc/article/view/60

[3] M. Alkoshman and S. Atiewi, "Empirical comparison of energy efficiency between 3-tier and high-speed 3-tier network topologies," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 13, no. 1, pp. 1–28, 2023. [Online]. Available: https://doi.org

[4] M. Vecchio, P. Azzoni, A. Menychtas, I. Maglogiannis, and A. Felfernig, "A fully open-source approach to intelligent edge computing: Agile's lesson," *Sensors*, vol. 21, no. 4, p. 1309, feb 2021. [Online]. Available: https://doi.org/10.3390/s21041309

[5] R. Mishra, S. Sharma, and A. Kumar, "Performance comparison of type-1 hypervisors in full hardware virtualization: ESXi, KVM, Hyper-V, and Xen," *Advances in Electrical and Computer Engineering*, vol. 21, no. 1, pp. 89–98, 2021. [Online]. Available: https://aece.ro/displaypdf.php?year=2021number=1article=2

[6] J. Hwang, S. Zeng, F. Y. Wu, and T. Wood, "A component-based performance comparison of four hypervisors," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, pp. 269–276.

[7] N. Kraljević, B. Bokić, V. Popović, and V. Golemović, "File system performance comparison with Proxmox VE and KVM as type-1 Linux-based hypervisors," in *2025 24th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, 2025, pp. 1–5.

[8] B. orević and V. Timčenko, "Performance comparison of KVM and Proxmox type-1 hypervisors," in *2022 11th International Conference on Zooming Innovation in Consumer Technologies (ZINC)*. IEEE, 2022, pp. 155–160.

[9] V. Sandeep and R. Davu, "Comparative analysis of leading HCI solutions: VMware vSAN vs. Azure Stack HCI vs. Nutanix," *Journal of Technological Innovations (JTI)*, vol. 5, no. 2, pp. 1–11, 2024.

[10] J. Tharwani, S. Aggarwal, and A. Purkayastha, "Quantifying virtualization overhead for cpu-intensive hpc workloads in hypervisors: A comparative study of kvm, proxmox ve, xen, and hyper-v," in *2026 IEEE SoutheastCon*. IEEE, 2026, pp. 1–6.

[11] B. orević, V. Timčenko, N. Kraljević, and N. Maček, "File system performance comparison with KVM and Xen as type-1 Linux-based hypervisors," in *Proceedings of the International Scientific Conference Sinteza*. Singidunum University, 2025, pp. 97–104.

[12] S. A. Algarni, M. R. Ikbal, R. Alroobaea, A. S. Ghiduk, and F. Nadeem, "Performance evaluation of Xen, KVM, and Proxmox hypervisors," *International Journal of Open Source Software and Processes (IJOSSP)*, vol. 9, no. 2, pp. 39–53, 2018.

[13] H. u. Rahman, G. Wang, J. Chen, and H. Jiang, "Performance evaluation of hypervisors and the effect of virtual CPU on performance," in *2018 IEEE 15th International Conference on Ubiquitous Intelligence and Computing (UIC)*. IEEE, 2018, pp. 1118–1125.

[14] C.-K. Tsung, J.-C. Liu, S.-H. Liu, V. K. Verma, and C.-T. Yang, "Performance analysis in hyperflex and vsan hyper convergence platforms for online course consideration," *IEEE Access*, vol. 10, pp. 124 464–124 474, 2022.