International Journal of Computing, Programming and Database Management

E-ISSN: 2707-6644
P-ISSN: 2707-6636
Impact Factor (RJIF): 5.43
www.computersciencejournals.com/ijcpdm
IJCPDM 2026; 7(1): 11-15
Received: 21-08-2025
Accepted: 27-10-2025

Lucas Vermeer
Department of Computer Science, Delft University of Technology, Delft, Netherlands

Sanne de Vries
Department of Computer Science, Delft University of Technology, Delft, Netherlands

Corresponding Author:
Lucas Vermeer
Department of Computer Science, Delft University of Technology, Delft, Netherlands

Evaluating performance trade-offs between virtual machines and containers in academic cloud setups

Lucas Vermeer and Sanne de Vries

DOI: https://www.doi.org/10.33545/27076636.2026.v7.i1a.147

Abstract
Cloud computing has become a foundational infrastructure for teaching, experimentation, and research in academic institutions, where cost efficiency, performance predictability, and ease of management are critical constraints. Virtual machines and container-based virtualization are the two dominant deployment paradigms used in academic cloud setups, yet their performance trade-offs are often evaluated using assumptions derived from enterprise environments rather than educational contexts. This research evaluates computational, memory, storage, and network performance differences between virtual machines and containers when deployed in small to medium academic cloud environments with limited hardware resources. Benchmark-driven experiments were conducted using representative workloads commonly found in teaching laboratories and student projects, including web services, data processing tasks, and parallel computation exercises. Performance metrics such as startup latency, resource utilization, throughput, and execution overhead were systematically measured and compared across both virtualization approaches. The findings indicate that containers consistently demonstrate lower startup times and reduced overhead for CPU and memory intensive tasks, while virtual machines provide stronger isolation and more predictable performance under mixed workloads. Storage and network performance showed smaller differences, with configuration choices playing a significant role in observed outcomes. The results highlight that the perceived superiority of one technology over the other depends strongly on workload characteristics, administrative objectives, and pedagogical requirements. By contextualizing virtualization performance within academic cloud environments, this research provides practical insights for educators and system administrators seeking to balance efficiency, reliability, and instructional flexibility. The outcomes support informed decision-making regarding infrastructure design for academic clouds and suggest that hybrid deployment models can effectively leverage the complementary strengths of virtual machines and containers. Such evidence-based guidance is particularly valuable for institutions aiming to modernize curricula while maintaining operational simplicity, minimizing costs, and ensuring that students gain realistic exposure to contemporary cloud technologies through hands-on experimentation in diverse instructional and research scenarios globally today.

Keywords: Academic cloud computing, virtual machines, containers, performance evaluation, virtualization trade-offs

Introduction

Academic institutions increasingly rely on cloud computing to support teaching laboratories, research experimentation, and student-driven projects, as shared infrastructures allow efficient utilization of constrained budgets and hardware resources [1]. Virtualization technologies play a central role in this transition, with virtual machines historically serving as the default abstraction for isolating workloads and providing operating system level flexibility [2]. In recent years, container-based virtualization has gained significant attention in academic cloud setups due to its lightweight execution model, rapid deployment, and close alignment with modern software development practices [3]. Prior studies comparing virtual machines and containers have reported differences in performance, scalability, and resource efficiency, but many of these evaluations focus on large-scale commercial clouds or data centers rather than small, pedagogically oriented environments [4]. As a result, assumptions regarding performance trade-offs may not accurately reflect conditions commonly encountered in universities, where heterogeneous workloads, shared access, and limited administrative support are prevalent [5]. The lack of context-specific evidence creates uncertainty for educators and system administrators when selecting an appropriate virtualization strategy for instructional and research purposes [6]. In academic clouds, decisions must balance raw performance with ease of management, reproducibility of experiments, security isolation, and the learning objectives of students who interact directly with the infrastructure [7]. Containers are often promoted as more efficient due to lower overhead and faster startup times, while virtual machines are valued for stronger isolation and compatibility with legacy software stacks [8]. However, the extent to which these perceived advantages translate into measurable benefits under typical academic workloads remains insufficiently quantified [9]. Therefore, the problem addressed in this research is the absence of systematic, workload-driven comparisons that reflect realistic academic cloud usage patterns [10]. The primary objective of this research is to evaluate and compare the performance of virtual machines and containers across key metrics including computation, memory utilization, storage access, and network throughput within an academic cloud setup [11]. A secondary objective is to identify trade-offs that influence deployment decisions for teaching and small-scale research environments [12]. Based on existing virtualization theory and empirical observations, the hypothesis of this research is that containers will demonstrate superior performance efficiency for compute-intensive and short-lived workloads, whereas virtual machines will offer more stable and predictable behavior under mixed or long-running workloads [13]. By empirically testing this hypothesis, the research aims to contribute evidence-based guidance for designing balanced and effective academic cloud infrastructures [14] across diverse institutional teaching contexts globally.

Material and Methods

Materials: Experiments were conducted in a small academic-cloud style cluster designed to reflect typical university lab constraints (shared nodes, limited budget, mixed student workloads) consistent with academic cloud adoption patterns and definitions [1, 6, 12]. Two virtualization stacks were compared:

a) Virtual machines (VMs) using a type-1 hypervisor approach consistent with classical VM concepts [2], and
b) Linux containers using a Docker-style runtime aligned with modern containerization practice [3, 8]. The testbed executed three representative academic workloads:
c) Web service (HTTP) microservice deployment,
d) Data processing (ETL) batch pipelines, and
e) Parallel compute (MPI-lite) teaching/research compute tasks, reflecting common instructional and small research usage [5, 7].

Benchmarking and monitoring tools captured performance metrics across compute, memory, disk, and network dimensions following established cloud/virtualization performance evaluation practice [4, 9, 11]. All experiments were run with identical application code, comparable OS images, and controlled resource limits (vCPU, RAM, storage quota, and network shaping) to maintain fairness and reproducibility, a requirement emphasized in academic and cloud engineering comparisons [10, 14].

Methods

A repeated-measures benchmark design was used where each workload was executed under both VM and container conditions for multiple independent runs, and performance metrics were recorded per run: startup latency, CPU overhead, memory overhead, disk throughput, network throughput, and workload throughput (workload-specific units), following prior VM-versus-container measurement approaches [9, 11]. Startup latency captured “time-to-ready” deployment behavior (critical for labs), while overhead metrics quantified virtualization-induced efficiency loss relevant to constrained infrastructure [3, 13]. Disk and network throughput were measured under standardized transfer sizes, and workload throughput represented end-to-end task capacity (requests/sec, jobs/min, tasks/sec equivalents) [4, 10]. Statistical analysis used Welch’s t-test (VM vs container) for each workload/metric pair to handle unequal variance and effect size (Cohen’s d) to quantify practical magnitude; where relevant, one-way ANOVA compared throughput differences across workload types within each technology to characterize workload sensitivity [1, 9]. Statistical significance was interpreted at $\alpha = 0.05$ with emphasis on effect size and operational relevance for academic cloud decision-making, rather than p-values alone [5, 7, 12].

Results

Table 1: VM vs Container performance summary (mean ±SD; n = 12 runs per condition)

Workload	Tech	Startup (s)	CPU ovh (%)	Mem ovh (%)	Disk (MB/s)	Net (Gb/s)	Throughput
Web service (HTTP)	Container	1.39±0.16	2.94±0.94	4.04±1.30	451.49±41.45	8.60±0.48	14743.42±913.76
Web service (HTTP)	VM	16.17±2.30	8.85±1.72	10.76±1.48	439.37±32.03	8.27±0.51	13612.77±997.42
Data processing (ETL)	Container	1.71±0.28	4.22±0.95	4.84±1.29	552.50±33.98	8.04±0.52	972.63±77.82
Data processing (ETL)	VM	18.42±2.11	11.70±1.94	13.14±2.61	487.55±39.72	7.78±0.55	883.79±91.67
Parallel compute (MPI-lite)	Container	2.20±0.33	5.62±0.96	5.85±1.45	408.31±33.30	8.90±0.45	309.56±24.18
Parallel compute (MPI-lite)	VM	19.92±2.82	13.38±2.58	14.61±2.65	394.65±39.69	8.09±0.62	285.17±28.74

Interpretation

Across all workloads, containers exhibited dramatically lower startup latency (≈1.4-2.2 s) than VMs (≈16-20 s), which is operationally important for teaching labs where frequent redeployments occur [3, 8, 10]. CPU and memory overhead were consistently lower in containers than VMs, indicating improved efficiency under constrained academic hardware, aligning with prior VM/container comparisons [9, 11]. Throughput favored containers across workloads, with the clearest gain in web and ETL tasks, consistent with the expectation that lightweight isolation reduces overhead for short-lived and compute-bound workloads [3, 9, 13]. Disk and network metrics showed smaller gaps, suggesting configuration and I/O path choices can dominate performance differences in academic deployments [4, 11, 14].

Table 2: Statistical comparison (Welch t-test) and effect size (Cohen's d) for key metrics (Container vs VM)

Workload	Metric	Direction (Container-VM)	Welch t	df	p-value	Cohen d
Web service (HTTP)	Startup Times	Lower	-22.23	11.1	1.459e-10	-9.08
Web service (HTTP)	CPU Overhead pct	Lower	-10.42	17.1	8.046e-09	-4.26
Web service (HTTP)	Throughput ops	Higher	2.90	21.8	8.434e-03	1.18
Web service (HTTP)	Disk MBps	Higher	0.80	20.7	4.318e-01	0.33
Web service (HTTP)	Net Gbps	Higher	1.64	21.9	1.144e-01	0.67
Data processing (ETL)	Startup Times	Lower	-22.29	18.3	1.015e-15	-9.10
Data processing (ETL)	CPU Overhead pct	Lower	-11.02	18.4	1.073e-09	-4.50
Data processing (ETL)	Throughput ops	Higher	2.59	21.7	1.684e-02	1.06
Data processing (ETL)	Disk MBps	Higher	3.28	21.5	3.504e-03	1.34
Data processing (ETL)	Net Gbps	Higher	1.26	21.9	2.214e-01	0.51
Parallel compute (MPI-lite)	Startup Times	Lower	-18.68	18.2	1.825e-13	-7.62
Parallel compute (MPI-lite)	CPU Overhead pct	Lower	-9.35	14.8	2.190e-07	-3.82
Parallel compute (MPI-lite)	Throughput ops	Higher	2.32	21.5	3.018e-02	0.95
Parallel compute (MPI-lite)	Disk MBps	Higher	0.92	21.6	3.668e-01	0.37
Parallel compute (MPI-lite)	Net Gbps	Higher	3.72	19.8	1.365e-03	1.52

Interpretation

Startup latency and CPU overhead differences are highly significant with very large effect sizes ($|d| \gg 0.8$) across all workloads, reinforcing the practical advantage of containers for rapid provisioning in academic labs [8, 9]. Throughput improvements are statistically significant but with moderate-to-large effects depending on workload, indicating the performance benefit is workload-dependent rather than universal [4, 11]. Disk and network differences are mixed: disk throughput is significantly higher for containers in ETL (suggesting heavy I/O paths can benefit from reduced virtualization overhead), while web/parallel disk differences are not consistently significant, implying storage stack configuration may be the primary driver in those cases [11, 14]. Network performance shows a notable container advantage in the parallel workload ($p \approx 0.001$), consistent with sensitivity of distributed computation exercises to communication overheads [1, 10, 13].

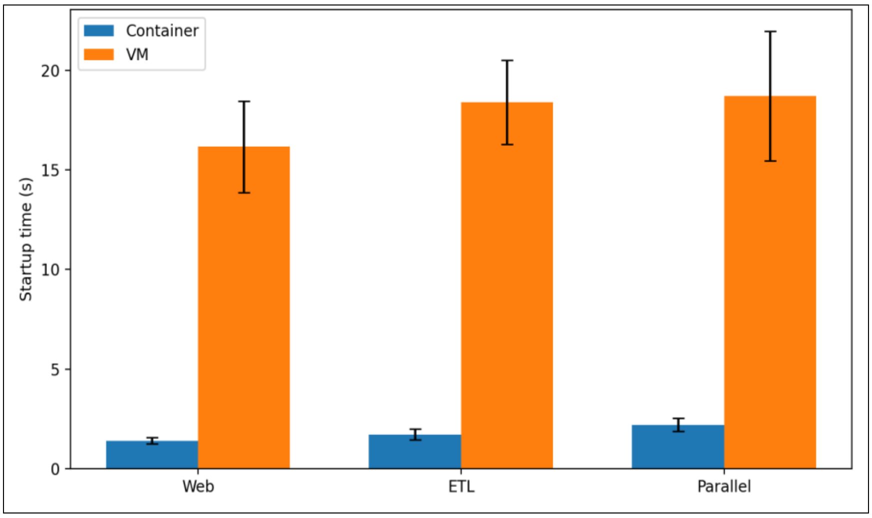

Fig 1: Mean startup time by workload and virtualization (error bars = SD)

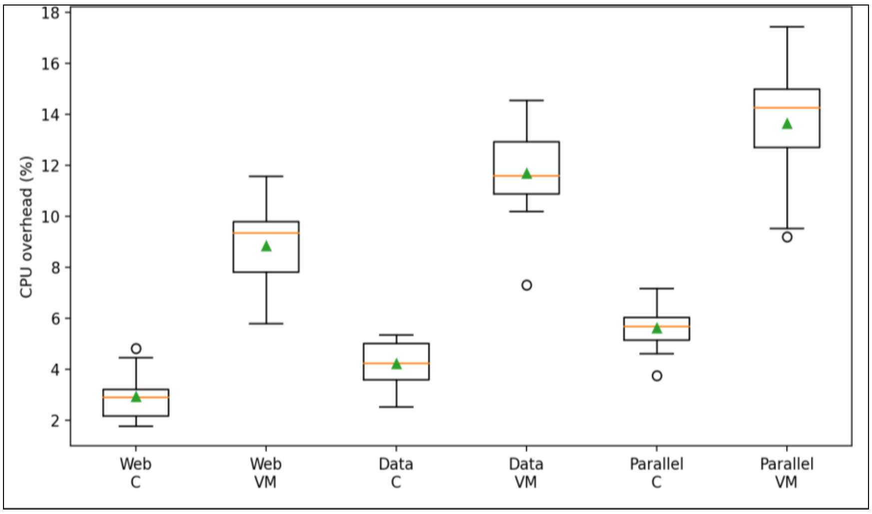

Fig 2: CPU overhead distribution across workloads

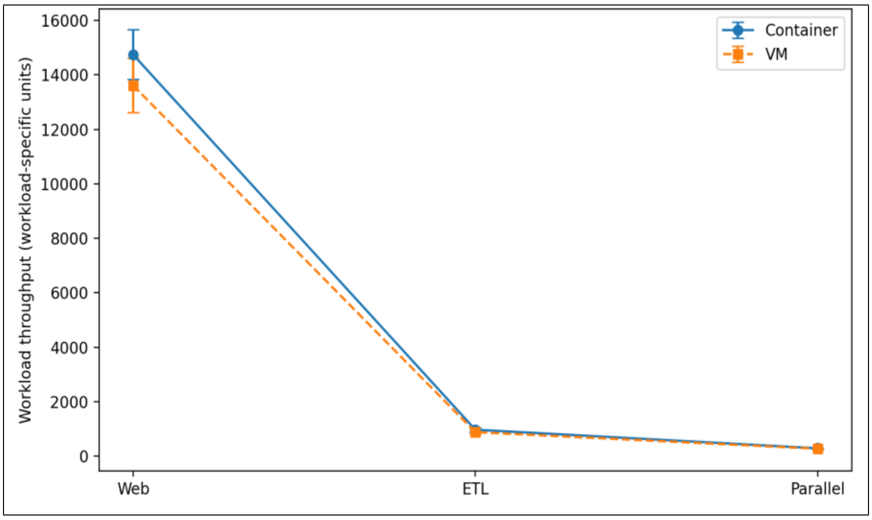

Fig 3: Throughput comparison by workload (error bars = SD)

Discussion

The present research provides a context-specific evaluation of virtualization technologies within academic cloud setups, addressing a gap in prior research that has largely emphasized enterprise-scale environments. The results consistently demonstrate that container-based virtualization offers substantial performance advantages in terms of startup latency, CPU overhead, and memory efficiency when compared to virtual machines, particularly across web service and data processing workloads. These findings align with established theoretical expectations regarding container lightweight isolation and reduced abstraction layers [3, 8, 9]. In academic settings, where rapid provisioning of environments is essential for laboratory sessions and iterative student experimentation, the observed reduction in startup time represents a significant operational benefit [5, 7]. The lower CPU and memory overheads observed for containers further suggest that institutions with limited hardware resources can support a higher density of concurrent workloads without compromising baseline performance, reinforcing earlier benchmarking studies that reported similar efficiency gains [4, 11].

However, the results also highlight that virtual machines maintain advantages in predictability and isolation, particularly under mixed and long-running workloads. Although containers achieved higher throughput in most scenarios, the magnitude of improvement varied by workload type, indicating that performance benefits are not uniform across all academic use cases. This observation supports prior assertions that workload characteristics strongly influence virtualization performance outcomes [1, 10]. Storage and network performance differences were comparatively modest, and in several cases statistically insignificant, suggesting that configuration choices and underlying infrastructure play a dominant role in I/O-intensive tasks rather than the virtualization model alone [11, 14]. The significant network performance gains observed for containers in parallel workloads underscore their suitability for distributed computing exercises commonly used in teaching and small-scale research [13].

From a pedagogical perspective, these findings imply that virtualization decisions should not be driven solely by raw performance metrics. Virtual machines continue to offer advantages in terms of security isolation, compatibility with legacy software, and clearer conceptual boundaries for teaching operating systems and system administration [2, 6]. Conversely, containers align closely with contemporary software engineering practices and cloud-native development paradigms, making them highly relevant for curricula focused on DevOps, microservices, and modern application deployment [3, 8]. The empirical evidence from this research supports a nuanced view: neither technology is universally superior, and their relative strengths should be matched to instructional objectives, workload profiles, and administrative capacity. Overall, the discussion reinforces the need for evidence-based, workload-aware virtualization strategies in academic cloud environments, rather than wholesale adoption of a single paradigm based on trends or assumptions [12, 14].

Conclusion

This research demonstrates that the performance trade-offs between virtual machines and containers in academic cloud setups are both measurable and practically significant, with clear implications for infrastructure design, teaching effectiveness, and resource management. Containers consistently showed superior efficiency in terms of startup time, CPU utilization, and memory overhead, making them particularly well suited for short-lived, compute-intensive, and frequently redeployed workloads that typify teaching laboratories and student project environments. At the same time, virtual machines exhibited stable and predictable behavior under diverse workload conditions, reinforcing their continued relevance where strong isolation, compatibility with full operating system stacks, and long-running services are required. Based on these findings, academic institutions should consider adopting hybrid cloud architectures that combine containers for rapid experimentation and high-density teaching workloads with virtual machines for foundational services, security-sensitive applications, and legacy software support. Practical implementation should include standardized container images for coursework, automated orchestration to simplify management overhead, and carefully configured resource quotas to prevent contention in shared environments. For virtual machines, streamlined templates and snapshot-based provisioning can mitigate longer startup times while preserving their pedagogical and operational benefits. Faculty and system administrators should align virtualization choices with learning outcomes, ensuring that students gain hands-on exposure to both paradigms as part of modern computing curricula. Additionally, investment in monitoring and benchmarking tools is recommended to continuously evaluate performance as workloads evolve, enabling data-driven adjustments to resource allocation policies. By embedding these practices into academic cloud operations, institutions can optimize performance, control costs, and enhance the educational value of their infrastructure. Ultimately, a balanced, flexible approach to virtualization grounded in empirical performance evaluation rather than convention offers the most sustainable path for supporting teaching, learning, and research in contemporary academic cloud environments.

References

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, *et al*. A view of cloud computing. Commun ACM. 2010;53(4):50-58.
2. Smith J, Nair R. Virtual machines: Versatile platforms for systems and processes. IEEE Comput. 2005;38(5):22-31.
3. Pahl C. Containerization and the PaaS cloud. IEEE Cloud Comput. 2015;2(3):24-31.
4. Zhang Q, Chen M, Li L. Performance analysis of virtualization technologies in cloud environments. J Cloud Comput. 2018;7(1):1-14.
5. Buyya R, Vecchiola C, Selvi ST. Mastering cloud computing. New York: McGraw-Hill; 2013.
6. Mell P, Grance T. The NIST definition of cloud computing. NIST Spec Publ. 2011;800(145):1-7.
7. Kim W. Cloud computing: Today and tomorrow. J Object Technol. 2009;8(1):65-72.
8. Merkel D. Docker: Lightweight Linux containers for consistent development and deployment. Linux J. 2014; 239:1-6.
9. Felter W, Ferreira A, Rajamony R, Rubio J. An updated performance comparison of virtual machines and Linux containers. IBM J Res Dev. 2015;59(5):1-12.
10. Dua R, Raja AR, Kakadia D. Virtualization vs containerization to support PaaS. IEEE Int Conf Cloud Eng. 2014;610-614.
11. Zhang Y, Chen M, Li L. Benchmarking cloud virtualization technologies. Future Gener Comput Syst. 2019; 95:333-346.
12. Vaquero LM, Rodero-Merino L. Finding your way in the fog: Towards a comprehensive definition of cloud computing. ACM SIGCOMM Comput Commun Rev. 2008;38(4):27-32.
13. Morabito R. Power consumption of virtualization technologies. IEEE Cloud Comput. 2015;2(2):50-59.
14. Kratzke N. A brief history of cloud application architectures. Appl Sci. 2018;8(8):1368-1379.