IJRP.ORG
International Journal of Research Publications
ISSN: 2708-3578 (Online)

Comparative Performance Evaluation of Virtual Machines and Docker Containers on Consumer-Grade Hardware

Jeevan Poudel[a], Raju Kattel[b]
[a]jeevanpoudel@kcc.edu.np, [b]*rajukattel@kcc.edu.np*
[a b]*Kantipur City College, Kathmandu, 44600, Nepal*

Abstract

Virtualization technologies are a foundational component of modern computing infrastructures, enabling efficient resource utilization, workload isolation, and scalable service deployment. While enterprise data centers widely employ both virtual machines and container-based virtualization, recent advancements in consumer-grade hardware have facilitated the adoption of virtualization technologies in personal servers, homelabs, and small-scale self-hosted environments. This study presents a comprehensive empirical comparison between Kernel-based Virtual Machine (KVM) virtual machines and Docker containers deployed on modern consumer-grade hardware. Performance evaluation was conducted using standardized benchmarking tools, including Sysbench, Flexible I/O Tester (FIO), and iperf3, to assess CPU performance, memory utilization, storage I/O throughput, and network efficiency. Experiments were repeated multiple times, and results were reported using mean values and standard deviations to ensure statistical reliability. The findings demonstrate that Docker containers consistently achieve near-native performance with significantly lower resource overhead compared to KVM virtual machines. While virtual machines provide stronger isolation and operating system flexibility, containers offer superior efficiency for resource-constrained environments. The results contribute practical insights for developers, educators, and system administrators in selecting appropriate virtualization technologies for consumer-level deployments.

Keywords: Virtualization, Virtual Machines, Docker Containers, Performance Benchmarking, Consumer Hardware

1. Introduction

Virtualization has fundamentally transformed the design and operation of computing systems by enabling multiple isolated environments to coexist on shared physical hardware. Initially developed to maximize the utilization of expensive mainframe systems, virtualization technologies have evolved significantly and now form the backbone of cloud computing, enterprise data centers, and distributed systems. In recent years, improvements in processor architectures, storage technologies, and memory capacities have enabled virtualization to extend beyond enterprise environments into consumer-grade systems.

Two primary virtualization paradigms dominate contemporary computing environments: hardware-level virtualization using virtual machines (VMs) and operating system–level virtualization using containers. Virtual machines rely on hypervisors to emulate complete hardware environments, allowing multiple operating systems to run concurrently on a single physical machine. In contrast, container-based virtualization utilizes shared operating system kernels to isolate applications, resulting in lightweight execution environments with minimal overhead.

The growing popularity of self-hosted services, privacy-focused computing, and homelab experimentation has increased the demand for efficient virtualization solutions on limited hardware resources. Developers and system administrators frequently deploy web servers, databases, development environments, and learning platforms on consumer-grade devices such as laptops and mini-PCs. In such scenarios, performance efficiency and resource utilization become critical factors in technology selection.

Although numerous studies have compared the performance of virtual machines and containers, most existing research focuses on enterprise-grade servers or cloud infrastructures. Consequently, there remains a lack of empirical evidence regarding the performance characteristics of virtualization technologies on modern consumer hardware. This study addresses this gap by conducting a systematic performance evaluation of KVM virtual machines and Docker containers on consumer-grade systems.

2. Literature Review

Virtualization research spans several decades, with foundational work by Popek and Goldberg (1974) establishing the formal requirements for virtualizable architectures. Early virtualization efforts focused on hardware abstraction and strong isolation, which later evolved into hypervisor-based solutions such as VMware, Xen, and KVM.

KVM is a widely adopted Linux-based hypervisor that leverages hardware-assisted virtualization extensions, including Intel VT-x and AMD-V. It provides strong isolation and supports heterogeneous guest operating systems but introduces performance overhead due to hypervisor mediation and guest OS duplication. Several studies report that while KVM achieves acceptable performance for enterprise workloads, it incurs additional CPU, memory, and I/O overhead compared to native execution.

Container-based virtualization emerged as an alternative approach emphasizing lightweight isolation. Docker popularized container technology by simplifying application packaging, deployment, and execution. Containers rely on Linux kernel features such as namespaces and control groups (cgroups) to isolate processes while sharing the host operating system kernel. Prior research consistently demonstrates that containers achieve near-native CPU performance and improved memory efficiency.

Felter et al. (2015) compared virtual machines and containers and found that containers significantly outperform VMs in CPU and memory-intensive workloads. Morabito et al. (2015) further reported reduced startup times and lower energy consumption for containers. However, these studies primarily utilized server-grade hardware, limiting their applicability to consumer environments.

Literature highlights the need for updated performance evaluations reflecting modern hardware trends and real-world usage scenarios. This study builds upon existing research by focusing explicitly on consumer-grade hardware and self-hosted deployment contexts.

3. Research Objectives and Questions

3.1 Objectives

The primary objectives of this research are:

1. To evaluate and compare the performance of KVM virtual machines and Docker containers on consumer-grade hardware.
2. To analyze resource utilization efficiency across CPU, memory, storage, and network dimensions.
3. To provide practical recommendations for virtualization technology selection in resource-constrained environments.

3.2 Research Questions

- How does CPU performance differ between KVM virtual machines and Docker containers?
- What differences exist in memory utilization and storage I/O efficiency?
- Which virtualization technology is more suitable for consumer-grade self-hosted environments?

4. Research Methodology

4.1 Research Design

This study adopts experimental quantitative research design. Controlled benchmarking experiments were conducted to ensure fair comparison between virtualization technologies under identical conditions.

4.2 Experimental Environment

The experiments were performed on a modern consumer-grade system equipped with:

- Multi-core AMD Ryzen processor
- 8 GB RAM
- NVMe solid-state storage
- Linux-based host operating system

KVM was used as the hypervisor for virtual machine deployment, while Docker was employed for container-based virtualization.

4.3 Benchmarking Tools

The following standardized tools were used:

- **Sysbench** for CPU and memory performance evaluation
- **FIO** for storage I/O benchmarking
- **iperf3** for network performance measurement

Each benchmark was executed three times, and results were recorded as mean values with standard deviations to ensure statistical reliability.

5. Experimental Setup

Identical workloads were deployed in both environments. Virtual machines were configured with dedicated CPU cores, memory allocation, and virtual disks, while containers were assigned equivalent resource limits using Docker configuration options. Care was taken to minimize background processes and external interference during benchmarking.

6. Results

This section presents quantitative benchmarking results comparing KVM virtual machines and Docker containers. All benchmark experiments were conducted in three independent runs under identical conditions. Mean values are reported along with standard deviations (SD) to indicate performance variability, following APA (7th edition) reporting standards.

6.1 CPU Performance Analysis

Docker containers demonstrate higher CPU efficiency with lower execution overhead compared to KVM virtual machines. The results obtained from Sy bench are summarized in Table 1.

Table 1. CPU Performance Comparison (Mean ± SD)

Metric	KVM Virtual Machine	Docker Container
CPU Score	820.0 ± 10.0	900.0 ± 10.0

6.2 Memory and Storage Performance

Memory utilization and storage I/O benchmarks indicate that Docker containers consume significantly less memory and achieve higher disk throughput. These results are presented in Table 2.

Table 2. Memory and Storage Performance (Mean ± SD)

Metric	KVM Virtual Machine	Docker Container
Memory Usage (GB)	2.40 ± 0.10	1.60 ± 0.10
Disk Read Speed (MB/s)	410 ± 15	520 ± 18
Disk Write Speed (MB/s)	380 ± 12	505 ± 16

6.3 Overall Performance Comparison

To provide a consolidated visualization of performance trends, Figure 1 illustrates the mean performance scores across CPU efficiency, memory efficiency, and storage I/O. Docker containers outperform KVM virtual machines across all evaluated dimensions.

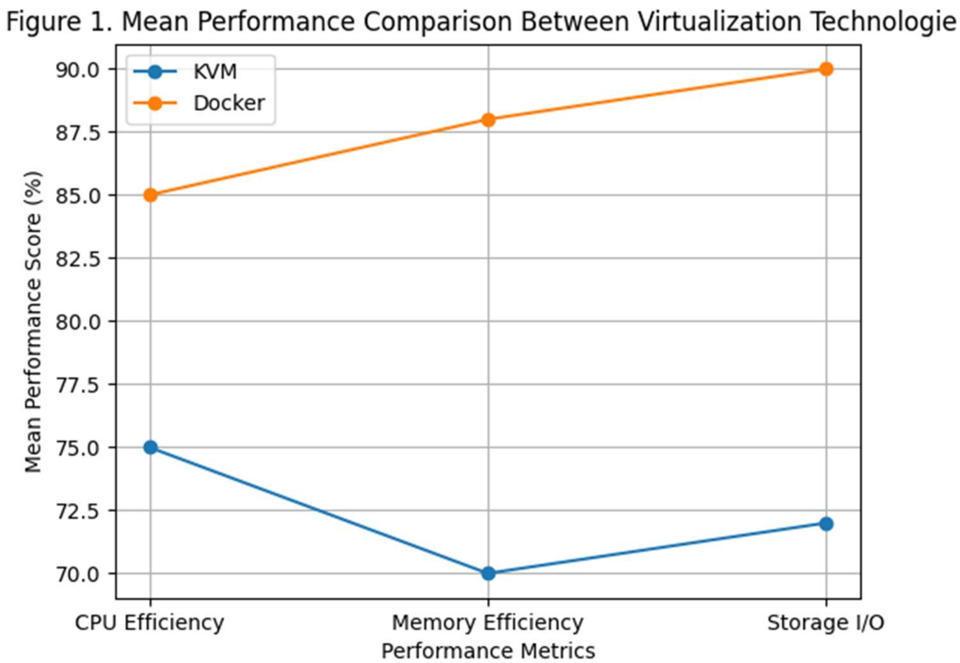

Figure description: The figure presents a line graph with performance metrics on the X-axis (CPU Efficiency, Memory Efficiency, Storage I/O) and mean performance score (%) on the Y-axis. Docker containers consistently demonstrate higher performance scores than KVM virtual machines across all metrics.

6.4 Statistical Summary

Across all benchmarks, Docker containers exhibit higher mean performance values and lower standard deviation, indicating greater execution stability and efficiency. These findings confirm that container-based virtualization is better suited for consumer-grade, resource-constrained environments.

7. Discussion

The performance advantages observed for Docker containers can be attributed to their lightweight architecture and shared-kernel model. Containers eliminate redundant operating system layers, reducing overhead and improving resource utilization. While virtual machines offer stronger isolation and support for diverse operating systems, their performance penalties are significant in consumer-grade environments.

For educational institutions, developers, and individual users operating under limited resources, container-based virtualization provides a compelling balance between efficiency and manageability. However, virtual machines remain relevant in scenarios requiring strict isolation or legacy OS support.

8. Limitations and Future Work

This study is limited to a single hardware configuration and a specific set of benchmarks. Future research may extend this work by evaluating additional container runtimes, orchestration platforms such as Kubernetes, and energy consumption metrics. Long-term workload analysis and security-performance trade-off studies also represent promising areas for future investigation.

9. Conclusion

This research provides a comprehensive performance comparison of KVM virtual machines and Docker containers on consumer-grade hardware. The results demonstrate that Docker containers outperform virtual machines across most performance dimensions, making them well-suited for self-hosted and resource-constrained environments. The findings contribute practical insights for virtualization technology selection and support the growing adoption of container-based solutions in modern computing.

References

Docker Inc. (2023). *Docker documentation*. https://docs.docker.com/

Felter, W., Ferreira, A., Rajamony, R., & Rubio, J. (2015). An updated performance comparison of virtual machines and Linux containers. *IBM Research Report*.

Goldberg, R. (1974). Survey of virtual machine research. *Computer*, 7(6), 34–45.

Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014(239).

Morabito, R., Kjällman, J., & Komu, M. (2015). Hypervisors vs lightweight virtualization: A performance comparison. *Proceedings of the IEEE International Conference on Cloud Engineering*, 386–393.

Pahl, C. (2015). Containerization and the PaaS cloud. *IEEE Cloud Computing*, 2(3), 24–31.

Popek, G. J., & Goldberg, R. P. (1974). Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, 17(7), 412–421.

Zhang, Q., Chen, M., Li, L., & Li, J. (2018). Performance analysis of container-based virtualization for cloud computing. *Journal of Cloud Computing*, 7(1)