Cloud Computing Report

Matteo Caiola
Università degli Studi di Trieste

7 Feb 2025

Background

This report is structured into distinct sections: the performance results for virtual machines (VMs) are presented first, followed by those of containers, concluding with a comparative analysis between the two approaches. The study highlights differences in performance, virtualization overhead, and the impact on resource allocation.

1 Introduction

This study evaluates VM and container performance in cloud computing using standardized tests. The test environment was configured to ensure equal resource allocation across both virtualization methods for an unbiased comparison.
Two **Ubuntu Server 24.04.1** virtual machines were deployed using **VirtualBox**[1], while two **Docker**[2] containers were launched with identical specifications. Each instance was allocated **2 CPUs and 2 GB of RAM**. The environments of VMs (**VM1** and **VM2**) and containers (**Container_1** and **Container_2**) were interconnected through an internal network. Furthermore, virtual machines were assigned **15 GB** of storage to ensure consistency and eliminate disk space as a performance factor, making CPU and memory utilization the primary variables in the analysis.

Both VMs were set up with two network adapters: Adapter 1 was configured as **NAT**, allowing internet access, while Adapter 2 was set to **internal network**, enabling direct communication between them. They were assigned static IPs: 192.168.56.2 and 192.168.56.3. To simplify remote management and testing, SSH access was enabled from the host system.
The two Docker containers were also created using the same **Docker Compose YAML configuration**. To ensure stable CPU performance, **core pinning** was used to bind container processes to specific CPU cores, reducing performance variations. They communicated through an internal **Docker network**, providing a controlled and isolated testing environment.

All tests were conducted on a **MacBook Air**, ensuring that no sequential paging was active during benchmark execution.

[1] https://github.com/Matterelloo/Cloud-Computing-Performance-Testing.git

The system specifications are as follows:

- **OS**: macOS Sonoma
- **CPU**: Apple M1 (8-core: 4 performance + 4 efficiency), Base/Boost clock speed: Variable (up to 3.2 GHz)
- **RAM**: 16GB unified LPDDR4X @4266 MT/s (shared with GPU)
- **Disk**: 512GB NVMe SSD

The benchmarking tools used were:

- **HPL**: Evaluates computing power by measuring FLOPS
- **Stress-ng**: Tests CPU and memory performance under load
- **IOZone**: Measures disk I/O speed for read/write operations
- **iperf**: Measures network throughput and performance between containers

These tools offer a comprehensive analysis of system performance. The complete benchmarking setup and results are available on **GitHub**[3] for reference and reproducibility.

2 Virtual Machines Performance Test

2.1 CPU Test - HPL

The performance of virtual machines in the HPL benchmark showed a maximum of 41.34 Gflops, with results ranging between 33.91 and 41.34 Gflops depending on the block configuration used. Given the theoretical optimal performance of 51.2 Gflops for a physical system with 2 CPUs at 3.2 GHz, the results indicate a measurable impact due to virtualization and resource management overhead.

2.2 General System Test: CPU & Memory Performance

The CPU and memory performance of virtual machines was assessed using stress-ng, with a workload of 2 CPUs and 1 GB of RAM for durations of 60 and 90 seconds. In CPU tests, VMs completed 24,255 operations in 60 seconds and 36,382 operations in 90 seconds, achieving an average processing rate of approximately 404 bogo ops/s. For memory operations, VMs processed 7,466,238 operations in 60 seconds and 13,410,318 in 90 seconds, with a peak throughput of 148,907 operations per second during the longer test.

2.3 Disk I/O Test

The disk I/O performance of virtual machines was evaluated using IOZone. The tests measured write, rewrite, read, and reread speeds for different block sizes. The results indicate that VMs perform well in sequential write operations, with peak speeds of up to 15.9 MB/s for 64 kB blocks and up to 34.3 MB/s for 256 kB blocks. Read operations generally outperformed write operations, reaching a maximum of approximately 64 MB/s. However, slight fluctuations were observed for smaller block sizes, indicating potential virtualization overhead affecting I/O performance. These results are summarized in Figure 1, which visualizes the impact of virtualization on disk I/O performance.

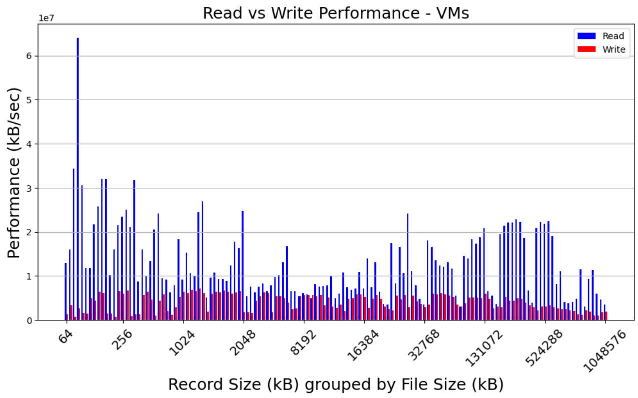

Figure 1: Disk I/O Performance of Virtual Machines

2.4 Network Test

Network performance tests on virtual machines were conducted using iperf3, measuring bandwidth in two scenarios: with a single stream and with two parallel streams (-P 2). During the 60-second test without -P 2, VMs achieved transfer speeds between 4.5 and 4.7 Gbps, but with a high number of retries and small size of the congestion window (Cwnd). When two parallel streams were introduced, the aggregate speed remained nearly the same, around 4.7 - 4.8 Gbps, while the number of retransmissions increased, indicating inefficiencies in handling multiple concurrent streams due to network virtualization overhead. These results highlight that virtualized network layers introduce significant performance limitations, particularly when dealing with multiple data streams.

3 Containers Performance Test

3.1 HPL Benchmark

In the HPL benchmark, containers outperformed VMs, achieving a maximum of 45.90 Gflops, with results ranging from 37.23 to 45.90 Gflops. This confirms that the absence of a hypervisor allows for more efficient resource utilization, bringing containers closer to the theoretical peak performance of 51.2 Gflops. These results further emphasize the computational efficiency of containerized environments, making them an ideal choice for high-performance workloads that require maximum processing power.

3.2 General System Test: CPU & Memory Performance

The CPU workload reached 24,279 operations in 60 seconds and 36,752 operations in 90 seconds, with an average processing speed of 408 bogo ops/s. For memory operations, containers executed 9,709,576 operations in 60 seconds and 15,662,972 operations in 90 seconds, surpassing VM performance with an average throughput of 174,028 operations per second.

3.3 Disk I/O Test

The disk I/O performance of containers was evaluated using IOZone. The tests measured write, rewrite, read, and reread speeds for different block sizes. The results indicate that containers achieve high sequential write speeds, reaching up to 19.5 MB/s for 64 kB blocks and up to 37.2 MB/s for 256 kB blocks. Read operations also showed strong performance, peaking at approximately 66 MB/s. While containers operate directly on the host system's storage without a hypervisor layer, they tend to have minimal I/O overhead compared to virtual machines. A detailed comparison of disk I/O performance for containers is presented in Figure 2, highlighting their efficiency advantages.

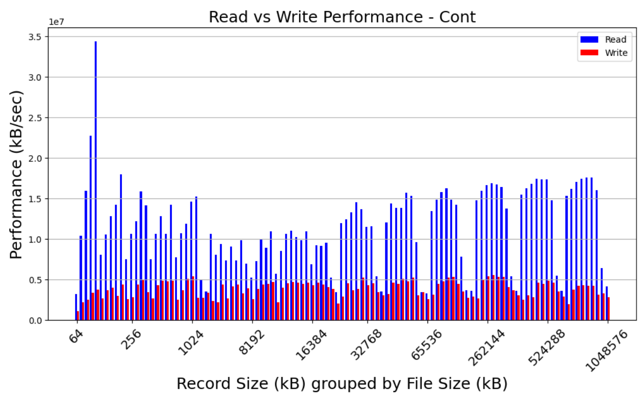

Figure 2: Disk I/O Performance of Containers

3.4 Network Test

In the single-stream test, containers achieved 62.5 Gbps, showing much more efficient internal communication, primarily due to the absence of a hypervisor and the direct use of the host network stack. With two parallel streams (-P 2), the total bandwidth further increased to 146 Gbps, demonstrating excellent scalability and a clear advantage in high-throughput network workloads. This drastic improvement highlights the efficiency of container networking, which utilizes direct host networking without the additional complexity of virtualized interfaces.

4 Comparison of Virtual Machine and Container Performance

4.1 CPU Test - HPL

The collected data clearly indicates that containers achieve superior computational efficiency compared to virtual machines in the HPL benchmark. While VMs reach commendable performance levels, they experience efficiency loss due to resource management by the hypervisor. Containers can better exploit the available computing power, narrowing the gap with the theoretical optimal value. The theoretical optimal performance of a system with these hardware specifications can be calculated using the following formula:

$$2 \times 3.2 \times 8 = 51.2 \text{ Gflops}$$

where:

- 2 represents the number of processors
- 3.2 GHz is the clock speed of each processor
- 8 are the FLOPS (floating-point operations) per second per cycle

Containers are therefore the preferred choice to maximize efficiency and computational power, as shown in Figure 3, demonstrating the efficiency gap between VMs and containers.

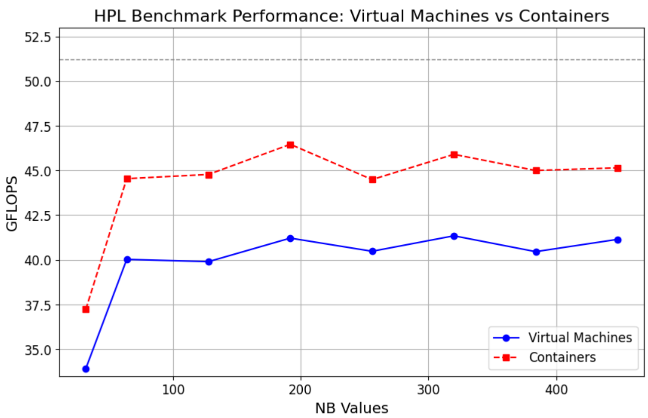

Figure 3: HPL Benchmark Performance

4.2 General System Test: CPU & Memory Performance

The results indicate that containers tend to deliver slightly better performance than VMs in both CPU and memory-intensive workloads. As mentioned above, this is due to the absence of a hypervisor, which reduces the overhead in resource management. However, the differences are not drastic, suggesting that both environments can be suitable for high CPU and memory utilization scenarios, with containers being more efficient for repetitive and long-running operations. The comparison of CPU and memory performance results is visualized in Figure 4, which highlights the efficiency gains observed in containerized environments.

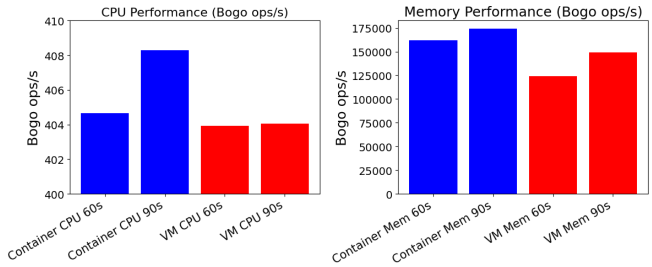

Figure 4: CPU and Memory Performance

I selected **bogo ops/s** as the primary metric among the options available, such as real-time, usr time, sys time, and bogo ops. This choice was made because **bogo ops/s** provides a direct measure of computational throughput in operations per second for CPU and memory workloads. Having focused on **bogo ops/s**, I ensure a consistent and comparable evaluation of performance across different environments.

4.3 Disk I/O Test

Containers generally get better performance than VMs in disk I/O operations, particularly in sequential read tasks. This improvement can be attributed to the absence of virtualization overhead introduced by a hypervisor. On the other hand, random read and write operations show no significant differences between the two environments, suggesting that filesystem management and host configuration play a dominant role. In general, containers are more efficient for workloads that require high-speed sequential data access, whereas virtual machines may still provide superior stability and isolation in more generalized usage scenarios.

4.4 Network Test

The comparative analysis clearly shows that containers significantly outperform VMs in network performance. Containers achieve more than 10 times higher transfer speeds compared to VMs in single-stream tests, with bandwidth reaching 62+ Gbps for containers versus only

4.7 Gbps for VMs. Figure 5 shows the efficiency of containerized networking even under basic conditions.

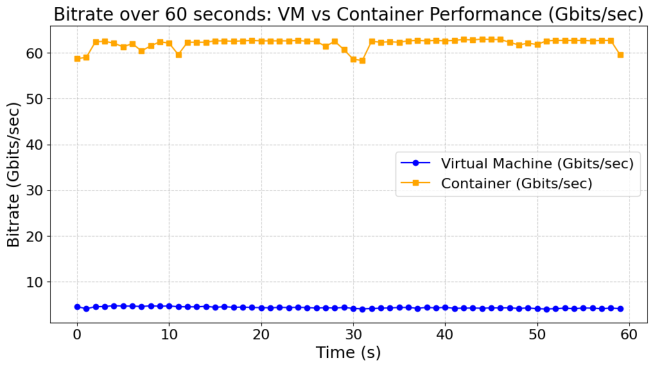

Figure 5: Single Stream

When two parallel streams are enabled (-P 2), the gap becomes even more pronounced. Containers exhibit excellent scalability, reaching up to 146 Gbps, while VMs remain limited to 4.8 Gbps. This significant performance difference highlights how container networking leverages direct access to the host network, avoiding the virtualized network stack overhead that constrains VMs. Figure 6 provides a clear visualization of this performance increase when parallel streams are introduced.

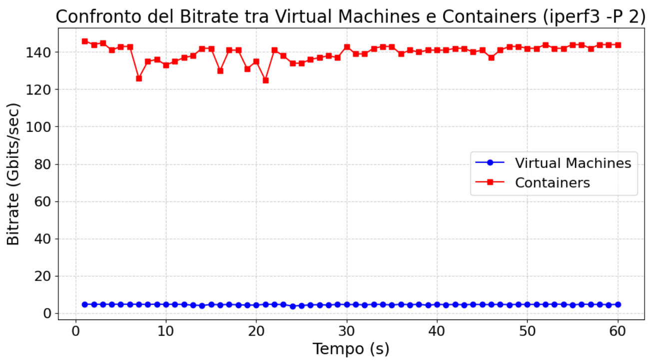

Figure 6: Parallel Streams

These results emphasize that while VMs may offer better isolation and structured resource management, containers provide a far superior networking performance, making them the preferred choice for high-bandwidth and low-latency applications.

Discussion & Conclusion

The analysis of the tests has highlighted significant differences in performance between virtual machines and containers. Containers have generally demonstrated superior efficiency in all executed tests, primarily due to reduced overhead compared to VMs. The impact of resource allocation and virtualization was observed in several key aspects:

- **Virtualization Overhead**: VMs exhibit less efficient resource management than containers due to the need for a hypervisor, which introduces latency in CPU and memory operations.
- **Filesystem Efficiency**: In Disk I/O tests, containers showed better performance in read and rewrite operations, demonstrating that direct resource management by the host positively influences I/O operations.
- **Network Performance**: In network tests, containers demonstrated significantly higher bandwidth compared to VMs due to the absence of an additional virtualization layer for network management.

This report demonstrates that, to achieve maximum performance in high-intensity computing scenarios, containers are preferable, while VMs remain a robust solution for environments requiring greater isolation and controlled resource virtualization.

Commands

- **HPL Benchmark:**
  - `./xhpl`
- **CPU Test:**
  - `stress-ng --cpu 2 --timeout 60/90s --metrics-brief`
- **Memory Test:**
  - `stress-ng --vm 2 --vm-bytes 1G --timeout 60/90s --metrics-brief`
- **Disk I/O Test:**
  - `iozone -a -g 1G -i 0 -i 1 2 -f`
- **Network Test (Single Stream):**
  - `iperf3 -s`

    `iperf3 -c 192.168.56.2 -t 60`
- **Network Test (Parallel Streams):**
  - `iperf3 -s`

    `iperf3 -c 192.168.56.2 -t 60 -P 2`

References

[1] *VirtualBox*. URL: https://www.virtualbox.org.

[2] *Docker*. URL: https://www.docker.com.

[3] Preston-Werner T. et al. *2008 GitHub: Let's build from here*. URL: https://github.com.