# Energy and Resource Modeling: A Comparative Analysis of Containers and Virtual Machines

**Donatas Dervinis**
Šiaulių valstybinė kolegija / Higher Education Institution

## Abstract

This paper presents a comparative analysis of energy and resource utilization between containers and virtual machines (VMs), technologies essential for modern cloud computing environments. Containers, lightweight virtualization solutions, enable rapid deployment, efficient scaling, and reduced overhead by sharing the host OS kernel, making them ideal for microservices and agile development workflows. Conversely, VMs offer enhanced security and isolation by virtualizing entire operating systems, suiting multi-tenant and legacy applications. Through mathematical modeling, this study quantifies the differences in energy consumption and resource efficiency of these technologies. The models utilize variables such as CPU and RAM usage and server load to assess each technology's performance in various scenarios. Results from simulations indicate that containers can significantly reduce infrastructure costs by optimizing resource allocation. A sample calculation for VMs and containers was performed to assess resource and energy demands. The results indicate that running 10 VMs requires 9.2% more CPU resources, and 12.5% more RAM compared to containers. In terms of energy consumption, VMs require 82% more energy than an equivalent setup of 10 containers.

**Keywords:** virtual machines, containers, resource, modeling, resource allocation**.**

## I. INTRODUCTION

In recent years containers and virtual machines (VMs) have gained increasing attention due to their critical roles in cloud computing, software deployment, and network function virtualization (NFV). Containers have risen in prominence due to their lightweight nature, offering advantages in performance, scalability, and resource efficiency compared to traditional VMs (Fig. 1) [1]. These technologies have become essential for deploying scalable and portable applications in various environments, from cloud infrastructure to edge computing.

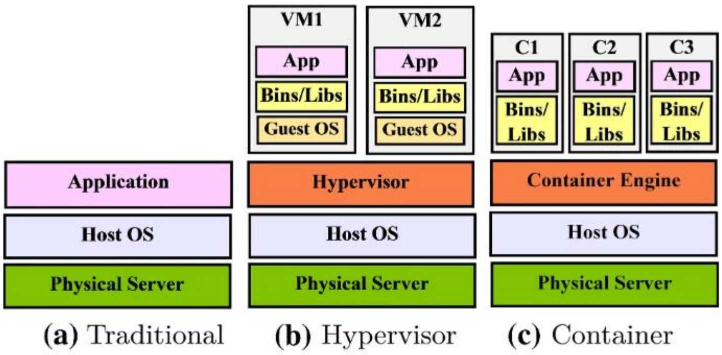

Fig 1. Comparison between application deployment using traditional hypervisor (virtual machine) and container architecture [1]

Containers, such as Docker, offer a more resource-efficient alternative to VMs because they share the host's operating system kernel, reducing overhead and enabling faster start-up times. This makes them ideal for microservices architectures and agile development cycles. However, while containers are lighter, VMs provide a higher level of isolation and security by virtualizing entire operating systems, making them more suitable for environments where strong security and isolation are critical, such as multi-tenant cloud environments [2].

Also show that containers offer better resource utilization, leading to lower infrastructure costs. The ability to spin up and down containers quickly in response to load variations makes them highly adaptable. According to Guerrero et al. (2021), containers provide more efficient resource optimization for microservices, reducing the need for additional hardware resources compared to VMs [3].

For instance, research shows that while VMs are highly secure, containers offer significant performance gains, particularly in scenarios requiring fast, scalable deployment [4]. Furthermore, container orchestration platforms like Kubernetes have accelerated the adoption of containers in cloud-native applications, offering sophisticated tools for managing container lifecycles, scaling, and resource optimization [5].

The relevance of both technologies continues to grow as industries move toward hybrid and multi-cloud environments. VMs are often used for legacy applications requiring greater security, whereas containers are favored for new, cloud-native development due to their agility and efficiency [6]. This trend is further driven by the increasing demand for energy-efficient solutions, with containers showing promise in reducing resource consumption in cloud data centers.

One of the primary concerns is the efficient scheduling and resource allocation of containers in Kubernetes clusters. Scheduling algorithms have evolved to handle dynamic workloads, with techniques such as topology-aware scheduling improving GPU resource utilization, while other strategies optimize CPU and memory usage across distributed systems. Studies highlight that Kubernetes' scheduling algorithms can be enhanced for better performance, like in edge computing scenarios where resource constraints are tighter [7]

**Task:** To develop models to systematically quantify the differences in performance and resource utilization between containerized environments and virtual machines

**Research goals:**

Develop energy and resource models.

Quantitatively evaluate the differences between containers and virtual machines.

**Methodology:** literature review, computer modeling.

## II. Energy and Resource Modeling

### *Development of Energy and Load Models*

When distributing resources in a containerized environment, we will construct a mathematical model of the energy system to maximize efficiency and minimize energy consumption. The notation used is as follows:

- $x_{ij}$ – Binary variable indicating whether container *i* is assigned to server *j* (1 – if yes, 0 – if no).
- *S* – count of servers.
- *C* – count of containers.
- *Pj* – Power consumption of server *j* in watts (energy consumption over time).
- $C_i$: – Resource requirements for container *i* (such as CPU, RAM, etc.).

First will describe server resource limitations:

$$\sum_{i=1}^{C} R_i \cdot x_{ij} \leq C_j, \nabla \mathrm{j} \in \{1,\ldots,\mathrm{S}\}, \tag{1}$$

where *Rj* is the resource capacity of server *j*

Each container *i* must be assigned to exactly one server *j*j:

$$\sum_{j=1}^{C} x_{ij} = 1, \nabla \mathrm{i} \in \{1,\ldots,\mathrm{C}\}. \tag{2}$$

Energy consumption can be calculated using the function:

$$min \sum_{j} P_j \cdot \left(\sum_{i} x_{ij} \cdot \frac{C_i}{R_j}\right). \tag{3}$$

Then the total energy consumption ($E_{total}$) can be described as:

$$E_{total} = \sum_{j=1}^{S} P_j \cdot \left(\sum_{i=1}^{C} \frac{R_i \cdot x_{ij}}{C_j}\right), \tag{4}$$

where (which is not mentioned above):

- $R_i$ – resource requirement of container $i$;
- $C_j$ – resource capacity of server j.

To measure performance, we will use the container request processing time and server load. This approach aims to maximize performance by minimizing the average request processing time. Where:

$T_i$ – Request processing time for container $i$.

$L_j$ – Load of server $j$.

Then, load balancing among servers $i$ can be described as:

$$\sum_i T_i \cdot x_{ij} \le L_j. \tag{5}$$

From equation (4), the performance will be equal to:

$$min \frac{1}{n} \sum_i T_i\,. \tag{6}$$

These equations enable the optimization of resource allocation in both containerized and virtual machine environments, aiming to reduce overall energy consumption. By using these formulas, companies can efficiently manage their infrastructure and reduce operational costs while maintaining high service quality.

## III. Comparison of Virtual Machines and Containers

Next, an example with 10 VMs containers will be calculated, demonstrating that containers are more efficient compared to virtual machines, a resource demand calculation will be performed. It is known that:

Virtual Machines: Virtual machines use a hypervisor, which creates a virtualization layer between the physical hardware and the operating system. This incurs additional resources overhead.

Containers: Containers share the host operating system kernel, avoiding this overhead and thus utilizing resources more efficiently.

Hardware Overhead: Hypervisors create virtual resources (CPU, memory, storage), which leads to additional overhead. Containers, on the other hand, share the OS resources, resulting in lower overhead.

Performance of Virtualization Layers: A hypervisor manages multiple VMs, which results in added costs due to context switching and hardware abstraction. Containers operate as isolated groups of processes without an additional virtualization layer, thereby offering higher performance.

According to both Microsoft Hyper-V [8] and VMware vSphere [9], the impact of the hypervisor on CPU and memory is minimal, often estimated at about 5% overhead.

**Resource Allocation:** If a VM requires 80% of resources [10] and considering a 5% overhead from the hypervisor [8,9], the effective resource utilization is approximately 85%.

**Energy Consumption:** Assuming that the server's energy consumption is proportional to resource usage, if the base power is $P_{base}$ and the maximum power at full load is $P_{max}$, energy consumption can be calculated as follows:

$$P_{VM} = P_{base} + 0.85 \cdot (P_{max} - P_{base}). \tag{7}$$

Meanwhile, containers have minimal overhead, typically around 2-3%, as they share the host OS kernel and do not require full OS installations.

Energy Consumption: For an 80% load with container workloads, the effective resource utilization is approximately 82% [11]. Then:

$$P_{Container} = P_{base} + 0.82 \cdot (P_{max} - P_{base}). \tag{8}$$

### Example resource calculation

Further calculations are performed using real parameters. We have a physical server with the following specifications:

- 16 physical cores (CPU);
- 64 GB RAM.
- 1000 GB SSD storage.
- Base power $P_{base}$=200 W.
- Maximum power $P_{max}$=600 W.

Each VM uses 1 GB of RAM for the OS and 0.1 CPU core. The VM hypervisor requires an additional 5% CPU overhead for managing all VM. Total count of VM – 10.

Containers share the main OS resources, which total use 2% of CPU and 2 GB of RAM.

***Effective resource usage for VM:***

$$\begin{gathered}\text{CPU: } 5\% \cdot\ 16\ (\text{Cores})\ = 0.8\ \text{Cores;}\\ \text{CPU of VM: } 10(VM) \cdot\ 0{,}1\ (\text{Core})\ = 1\ \text{Cores;}\\ \text{RAM: } 1\text{GB/each} \cdot 10(\text{VM}) = 10\ \text{GB.}\end{gathered} \tag{9}$$

Free resources available (based (9)):

$$\begin{gathered}\text{CPU: } 16 - 0.8 - 1 = 14{,}2\ Cores\\ \text{RAM: } 64\ -\ 10 = 54\ \text{GB.}\end{gathered} \tag{10}$$

Efficiency percentages with VM:

$$\begin{gathered}\text{CPU: } \frac{16-14{,}2}{16} \cdot 100 = 11.25\%;\\ \text{RAM: } \frac{64-54}{64} \cdot 100 = 15.6\%.\end{gathered} \tag{11}$$

***Calculations using containers:***

Main OS overhead:

$$\begin{gathered}\text{CPU: } 2\%(\text{see above}) \cdot\ 16\ (\text{Cores})\ = 0.32\ \text{Cores;}\\ \text{RAM: used by the main OS (see above): 2 GB}\end{gathered} \tag{12}$$

Available free resources:

$$\begin{gathered}\text{CPU: } 16 - 0.32\ = 15.68\ Core\\ \text{RAM: } 64\ -\ 2 = 62\text{GB.}\end{gathered} \tag{13}$$

Efficiency percentages with containers:

$$\begin{gathered}\text{CPU: } \frac{16-15{,}68}{16} \cdot 100 = 2\%;\\ \text{RAM: } \frac{64-62}{64} \cdot 100 = 3.1\%.\end{gathered} \tag{14}$$

The efficiency differences for a 10-unit run can be calculated based results (11) and (14):

$$\begin{gathered}\text{CPU: } 11{,}25\% - 2\% = 9{,}2\%;\\ \text{RAM: } 15{,}6\% - 3{,}1\% = 12{,}5\%.\end{gathered} \tag{15}$$

### *Calculation of energy consumption*

To evaluate the energy consumption for the previously calculated 10 VM and 10 containers, we will use the next average assumptions:

- CPU power consumption: 40 Watts per Core.
- RAM power consumption: 2,5 Watts per GB.

Energy consumption for 10 VM:

$$\begin{gathered}\text{CPU: } 1.8\ (Cores)\ \cdot\ 40W = 72W;\\ \text{RAM: } 10\text{GB} \cdot 2.5W = 25\text{W};\\ \text{Total: } 72{+}25\text{W}{=}97\text{W}.\end{gathered} \tag{16}$$

Energy consumption for 10 containers:

$$\begin{gathered}\text{CPU: } 0.32\ (Cores)\ \cdot\ 40W = 17.8W;\\ \text{RAM: } 2\text{GB} \cdot 2.5W = 5\text{W};\\ \text{Total: } 17.8\text{W}{+}5\text{W}{=}17.8\text{W}.\end{gathered} \tag{17}$$

The energetic efficiency differences for a 10-unit run can be calculated based on results (16) and (17):

$$97\text{W} - 17{,}8W = 79{,}2W, \tag{18}$$

i.e. 82% of VMs need more energy than 10 containers.

It is important to note that this is a model calculation that may vary depending on the specific load of the infrastructure. In a real environment, other factors such as network usage, I/O operations, and additional system overhead can affect overall energy consumption. These calculations compare only the system support resources (i.e., they do not include what will be run within the VMs or containers).

## CONCLUSION

A methodology for energy and resource calculation has been developed for the modeling of VM and container systems. Using these models, companies can estimate resource and energy requirements. A sample calculation for VMs and containers was performed to assess resource and energy demands. The results indicate that running 10 VMs requires 9.2% more CPU resources, and 12.5% more RAM compared to containers. In terms of energy consumption, VMs require 82% more energy than an equivalent setup of 10 containers.

## References

1. Bhardwaj A., Krishna C.R. *Virtualization in Cloud Computing: Moving from Hypervisor to Containerization—A Survey, Arabian Journal for Science and Engineering, Springer, 2021.*

2.Sultan S., Ahmad I. Container security: issues, challenges, and the road ahead. IEEE Xplore, 2019

3 Bentaleb Q, Belloum A. Containerization technologies: taxonomies, applications and challenges The Journal of Supercomputing, Springer, 2022.

4. Masdari, M., Zangakani, M. Green cloud computing using proactive virtual machine placement: challenges and issues. Journal of Grid Computing, Springer, 2020.

5. Guerrero C., Lera I., Juiz C. Resource optimization of container orchestration: a case study in multi-cloud microservices-based applications. Journal of Supercomputing, Springer, 2018.

6. Morabito R., Kjällman J., & Komu M. Hypervisors vs. lightweight virtualization: a performance comparison. IEEE International Conference on Cloud Engineering, 2015.

7. Turin G., Borgarelli A. Predicting resource consumption of Kubernetes container systems using resource models. Journal of Systems and Software, Elsevier, 2023.

8. Microsoft Virtual Machine Resource Controls Link: https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/manage/manage-hyper-v-cpugroups

9. H. Fayyad-Kazan, M. Timmerman. Benchmarking the Performance of Microsoft Hyper-V server, VMware ESXi and Xen Hypervisor. Journal of Emerging Trends in Computing and Information Sciences, 2013.

10. Reduce IT Costs and Increase Control with Server Virtualization. Link: https://www.vmware.com/solutions/consolidation.html.

11. D. Cao. Understanding CPU Resources and Memory Resources for Pod in Kubernetes. Links: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/.

**Received**: 4 November 2024.

**Accepted**: 27 November 2024