Journal of Grid Computing (2024) 22:67
https://doi.org/10.1007/s10723-024-09782-2

RESEARCH

The Goodness of Nesting Containers in Virtual Machines for Server Consolidation

Belen Bermejo · Carlos Juiz · Maria Carla Calzarossa

Received: 3 June 2024 / Accepted: 7 October 2024

Abstract Virtualization and server consolidation are the technologies that govern today's data centers, allowing both efficient management at the functionality level as well as at the energy and performance levels. There are two main ways to virtualize either using virtual machines or containers. Both have a series of characteristics and applications, sometimes being not compatible with each other. Not to lose the advantages of each of them, there is a trend to load data centers by nesting containers in virtual machines. Although there are good experiences at a functional level, the performance and energy consumption trade-off of these solutions is not completely clear. Therefore, it is necessary to study how this new trend affects both energy consumption and performance. In this work, we present an experimental study aimed to investigate the behavior of nesting containers in virtual machines while executing CPU-intensive workloads. Our objective is to understand what performance and energy nesting configurations are equivalent or not. In this way, administrators will be able to manage their data centers more efficiently.

Keywords Nesting · Consolidation · Virtual machines · Containers · Performance-energy trade-off · Monitoring · CPU-intensive Workloads

1 Introduction

Cloud Computing was introduced as a paradigm focused on offering services in a more flexible way for users, thus it has been necessary to provide cloud data centers with technologies such as virtualization. Through virtualization, cloud services can be deployed using different models, such as IaaS, PaaS, and SaaS [1].

Virtualization can be provided through virtual machines (and this happened since cloud computing inception) and more recently through containers. Unlike containers, virtual machines are more robust at the architectural level. In fact, virtual machines allow complete isolation between them, so that different systems can be deployed within the same physical server. Containers are processes within a physical server sharing its operating system, thus, their isolation factor is minimal compared to virtual machines [2]. Therefore, containers allow much faster deployment than virtual

Carlos Juiz and Maria Carla Calzarossa These authors contributed equally to this work.

B. Bermejo (✉) · C. Juiz
Computer Science Department, Universitat de les Illes Balears, Palma, Spain
e-mail: belen.bermejo@uib.es

C. Juiz
e-mail: cjuiz@uib.es

M. C. Calzarossa
Department of Electrical, Computer and Biomedical Engineering, Universitá di Pavia, Pavia, Italy
e-mail: mcc@unipv.it

Published online: 18 October 2024

machines. For example, a virtual machine can take minutes to be deployed, while a container can be deployed in a few seconds. This functional advantage often overshadows the isolation capacity of virtual machines.

To take advantage of the isolation of virtual machines and the fast deployment of containers, both technologies can be used at the same time, by hosting containers within virtual machines running on a given physical server. In fact, the current trend of cloud providers is the combined use of virtual machines and containers [3, 4]. This trend ensures security, isolation of the processes and faster deployment of services while maintaining the virtualization framework.

Since virtual machines and containers have different nature and aims, it is important to study the suitability (or the goodness) of nesting containers in virtual machines from performance and energy consumption simultaneously.

It is important to note that server consolidation introduces performance overhead [5, 6], and consequent performance degradation, that is, the response time increases and the QoS is affected negatively. Besides, the energy consumption depends directly on the performance (i.e., response time), and it is also affected by the performance consolidation overhead. For this reason, server consolidation may not save energy even though it saves power. Hence, it is necessary to find a trade-off between the performance degradation and the energy consumption [7]. As a consequence, the research question we attempt to answer in this paper is:

RQ: Is the current trend of nesting containers in virtual machines efficient from performance and energy consumption perspectives simultaneously?

To answer this question, we divided this paper into the following sections. In Section 2 some background concepts are introduced. In Section 3, the justification of this work is explained by the related work. Then, in Section 4 we describe the methodology followed to answer the research question. In Sections 5 and 6, the experimental results are presented and discussed in Section 7. We conclude this work in Section 8 with some final remarks.

2 Background

Since virtualization technology is a key factor in cloud data centers, in this section we will explore the concepts needed to understand the proposed work.

2.1 Virtualization

Although virtualization is an old concept, introduced by IBM on its mainframes more than fifty years ago, nowadays this disruptive technology has changed the way of handling a physical server and service delivery. Virtualization can be defined as a technology that divides the computing resources of a system into several isolated operating environments.

A Virtual Machine Manager (VMM), or hypervisor, is a software layer located between virtual machines and the hardware that manages the interactions between virtual machines (or containers) and the shared hardware. Hypervisors provide an environment identical to the physical one with minimal performance cost and complete control of the system's resources [6].

2.2 Virtual Machine-Based Virtualization

Virtual machine-based virtualization allows the execution of several isolated operating systems on the same physical server. These different operating systems are the virtual machines (VM). The Virtual Machine Manager allows communications of the virtual machines with the physical hardware, using the host operating system.

There are two types of hypervisors: type-I and type-II (see Fig. 1). Type-I (native or bare-metal) hypervisors run directly on top of the hardware, that is, they are placed on the operating system and interact directly with the Instruction Set Architecture interface. In contrast, type-II (or hosted) hypervisors need to be sup-

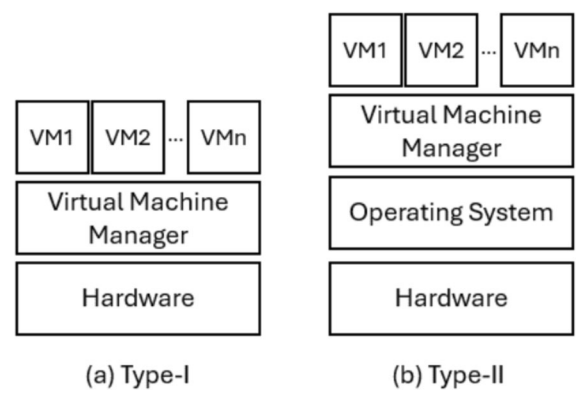

Fig. 1 Layered representation of two types of hypervisor (a) native or Type-I and (b) hosted or Type-II

ported by an operating system providing virtualization services. Hence, a type-II hypervisor is a program managed by the operating system.

From the performance point of view, type-I hypervisors are more efficient than type-II because they directly communicate with hardware resources in the stack below. In type-II hypervisors, every time a virtual machine operates, it has to hand off the requests to the operating system, which handles the hardware requests. Thus, type-II hypervisors need an extra software layer [5].

Type-I hypervisors are considered more secure than type-II hypervisors. Operations performed by the guest (i.e., virtual machine) are handed off, and as such, a guest cannot affect the hypervisor on which it is supported. A VM can affect only itself. For example, a guest crash does not leave the boundaries of the VM. In addition, type-II hypervisors are less reliable because there are multiple points of failure: anything that affects the availability of the underlying operating system can also impact the hypervisor and the guest it supports.

2.3 Container-Based Virtualization

Despite the recent use of containers, they were not that new in UNIX systems. In fact, since the end of the 1970s the `chroot` environments could be seen as the beginning of the containerization idea. We might consider the chroot idea as the first idea that very crudely separated processes from the disk space. At the beginning of this century, we had BSD jails, and in 2008 the LXC (Linux Containers) project was introduced to the market [8].

A container is a standard unit of software that packages codes and all their dependencies, so applications run quickly and reliably in different computing environments. Containers save resources because they avoid the overhead of virtualization, while also providing some isolation.

Table 1 compares virtual machines and containers on multiple factors such as performance isolation, security, networking and storage.

Application containerization is one of the technologies that enable microservices architectures. Microservices are organized around different business capabilities and requirements, thus their nature varies. Some microservices focus on data processing , others on data storage, and some others on processing HTTP requests and responses [9]. For this work, microservices are seen from the processing point of view, that is, the service time (small by definition) is devoted to CPU operations [2].

Containerization is a form of virtualization where applications run in isolated user spaces (containers), while using the same shared operating system (see Fig. 2). Therefore, each microservice of an application is deployed in a container, without launching a full hypervisor. Deploying a microservice is as simple as starting the execution of a new microservice container.

Table 1 Virtual machines and containers feature comparison

Parameter	Virtual machines	Containers
Operating system	Each VM runs its operating system.	Containers share the host operating system.
Communications (between VMs and containers)	Communications over the network, such as Internet.	Standard inter-process communication mechanisms like signals, pipes and sockets.
Security	It depends on the hypervisor implementation.	User control access can be leveraged.
Performance	Performance overhead due to the hypervisor layer (extra software layer).	Near-native performance as compared to the underlying host OS.
Isolation	Full since libraries and files cannot be shared between guests and between guest hosts.	Limited since subdirectories can be transparently mounted and shared.
Startup time	Minutes to boot.	A few seconds to boot.
Storage	Large amount of storage as the whole OS kernel and all programs have to be installed.	Low amount of storage as the base OS is shared.

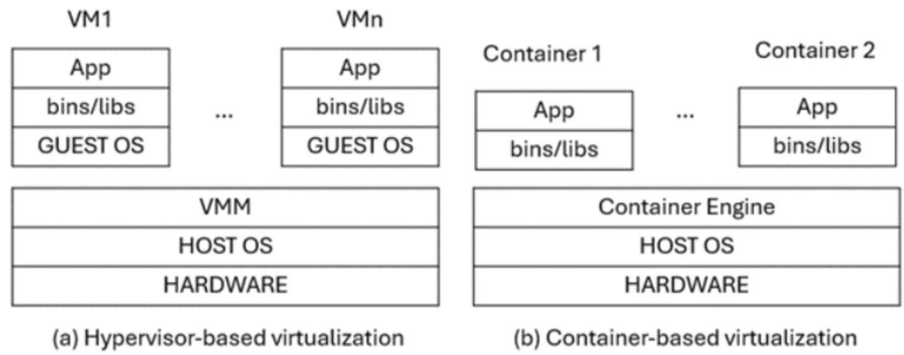

Fig. 2 Layered representations of (a) Hypervisor-based and (b) Container-based virtualization

Therefore, microservices can be scaled by simply creating new containers until the desired level of scalability is achieved. In this way, the overhead is reduced, and the isolation of the environments is maintained, although ensuring security is more difficult than for VMs. In fact, various authors claim that "containers do not contain" [10]. Docker [6] is one of the most successful implementations of container architectures.

2.4 Server Consolidation

Server consolidation is one of the techniques that help system administrators to manage servers and data centers more flexibly. The basic principle of this technique is the allocation of the workload in the minimum number of physical servers. The common way to consolidate servers is using virtual machine consolidation, which is based on the reallocation of virtual machines among different physical servers. For example, in Fig. 3 we represent three physical servers, each allocating one or more virtual machines. In this scenario, virtual machines allocated to server A and server C are migrated to server B (under the assumption that this server has enough resources). Hence, servers A and C can be switched off saving power [6].

The overhead of virtual machine consolidation is defined as the extra load that the physical server has to perform due to virtual machine management and coordination of the access to physical resources from the virtual instances (including resource contention). Consequently, the larger the number of consolidated virtual machines within the same physical machine, the higher the overhead is. Therefore, performance degradation occurs due to the overhead of consolidating virtual machines into physical machines [11].

In this work, we assume that server consolidation is performed in blocks of virtual machines and containers, that is, the consolidated virtual machines have a set of allocated containers. Hence, the inherent functional advantages of nesting containers in virtual machines will not receive any benefit from consolidation. In fact,

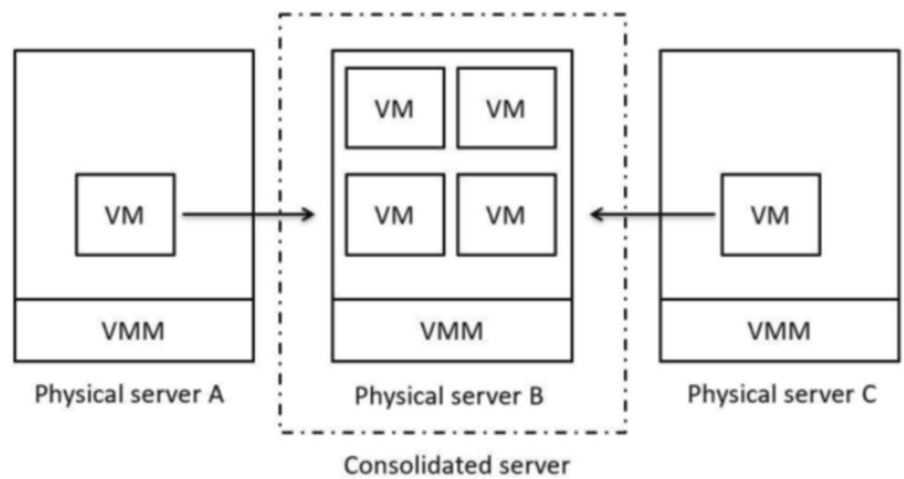

Fig. 3 Virtual Machine consolidation example

the saving in the power consumed by turning servers off will not correspond to saving in energy, which directly depends on performance. Therefore, if performance degrades, so the energy consumed does. Moreover, we assume CPU-intensive workloads able to stress the computing power of the physical servers.

3 Related Work

Since the rise of virtual machines and containers in data centers, many research works have emerged to evaluate their performance and energy consumption. In particular, several researchers investigated the performance of virtual machines and containers in data centers, whereas the nesting of different combinations of virtual machines and containers has been investigated by few researchers. In Table 2, we present a summary that compares the state of the art in this area. Also, we discuss the most relevant issues of these works.

In Shah et al. [12], authors studied the performance of different configurations of virtual machines and containers for cloud-based scientific workloads. The results obtained by running the HEPSCPEC06 scientific benchmark showed that modifications in the virtual machines and containers configurations, such as hyperthreading, isolation of CPU cores and allocation of vCPUs, can improve the throughput and performance of virtual machines and containers. Similarly, in Xavier et al. [13] authors performed a large number of experiments to analyze container-based virtualization for high performance computing environments in terms of performance and isolation. The experimental results confirmed that containers have the worse isolation due to the shared operating system and the worse performance for HPC workloads in comparison with virtual machines.

In Seo et al. [14], authors compared the performance of Linux Containers and Virtual Machine hypervisors by analyzing the boot speed and CPU performance, thus helping users in selecting the most suitable platform to deploy cloud microservices. Another study Li et al. [15] compared virtual machines and containers to assess the performance differences between these two types of virtualization solutions. From the experimental results, authors concluded that containers did not achieve higher performance because of their lower storage transaction speed. In Ruan et al. [16] authors measured the performance of application and system containers and concluded that system containers are more suitable to sustain I/O-bound workload than application containers.

As already mentioned, despite its importance, the nesting of containers in virtual machines has been researched to a rather limited extent. In Mavridis et al. [17], authors studied the combination of virtual

Table 2 Comparison with the state of the art

Paper	VM	Container	Consolidation/ Nesting	Energy	Performance	Energy-performance trade-off
[7]	X	–	Consolidation	X	X	X
[10]	X	X	Nesting	–	X	–
[11]	X	–	Consolidation	–	X	–
[12]	X	X	Consolidation	–	X	–
[13]	-	X	Consolidation	–	X	–
[14]	X	X	Consolidation	–	X	–
[15]	X	X	Consolidation	–	X	–
[16]	-	X	Consolidation	–	X	–
[17]	X	X	Nesting	X	X	-
[18]	X	X	Nesting	X	X	–
[19]	X	X	Nesting	–	X	–
[20]	X	X	Nesting	–	X	–
[21]	–	X	Consolidation	X	X	–
This work	X	X	Consolidation/nesting	X	X	X

machines and containers using different technologies such as KVM (Kernel-based Virtual Machines), Xen and Docker. Through experimentation, they measured the overhead associated with this combination and the impact of energy consumption. Unlike our work, this work does not consider performance and energy consumption at the same time.

In Shah et al. [18], authors evaluated the efficiency of different configurations of containers combined with virtual machines in OpenStack environments for running the HEPSCPEC06 benchmark. The focus of this work is on the power consumption rather than on energy consumption.

In Barik et al. [19], authors explored hosting containers and virtual machines in physical servers. They evaluated the overhead and the efficiency for different workloads (e.g., CPU, memory, I/O and SSL protocol). This work considered the combination of containers and virtual machines, although the energy consumption is not analyzed. In Mavridis and Karatza [20] authors evaluated the performance when deploying a container on the top of a virtual machine and concluded that the container performance is penalized by the additional virtualization layer of the virtual machine.

In Cuadrado-Cordero et al. [21] authors compared the containers consolidation with a virtual machine consolidation scenario from the QoS and energy efficiency point of view. Even though these metrics are relevant, authors did not consider them simultaneously. The QoS was also assessed in Bermejo and Juiz [11], where authors developed a general method to evaluate the performance overhead of virtual machines consolidation. However, the energy consumption was not considered.

Since containers are devoted to microservices, in Raza et al. [10] authors evaluated the performance of the combination of containers with virtual machines in a microservice IoT environment using real experimentation. Even this paper did not consider energy consumption.

Regarding the performance and the energy consumption simultaneously, in Juiz and Bermejo [7], authors performed an evaluation of the performance and energy consumption trade-off using the CiS^2, a metric that quantifies and represents the trade-off for any consolidation environment. This work evaluated the trade-off for different virtual machine-based hypervisors, but it did not consider containers.

From the analysed literature, to the best of our knowledge, our paper is the first attempt to study the benefits of nesting containers in virtual machines by assessing simultaneously performance and energy consumption.

4 Methodology

As previously stated, this work aims to study the goodness of nesting virtual machines and containers. To achieve this aim and answer our research question, we follow the classical performance engineering methods and techniques [22], specifically benchmarking and monitoring applied to real systems.

In detail, our methodology consists of several steps, namely:

- definition of characteristics of the workload, i.e., the benchmarks;
- generation of the workload on the real infrastructure, i.e., the System Under Test (SUT), chosen as target;
- monitoring the performance and behavior of the SUT under the identified workload.

Note that all these steps are driven by the objectives of the study. For example, this means that benchmarks representative of the scenarios being considered are to be chosen. For our study, these benchmarks will stress the SUTs by exercising their most power-demanding components, that is, their CPUs. Similarly, it is necessary to use hardware and software monitors able to accurately collect the identified attributes, that is, response time, power consumption and energy consumption.

We outline that measurements being collected at run time could be analyzed online, i.e., while the SUT is processing the benchmarks, or offline, that is, for a post-mortem analysis. In our study, we will perform an offline analysis to reduce intrusiveness. It is also worth mentioning that to ensure measurement quality multiple runs of the benchmarks on the SUTs have to be performed.

In what follows, we detail the experimental set-up by describing the main hardware and software character-

istics of our SUTs as well as the types of the workloads selected for assessing the goodness of various nesting configurations. We will also present the design of the experiments.

4.1 Experimental Set-Up

Three physical servers, i.e., hosts, have been used in the experiments. These SUTs have the following hardware features:

- Lenovo ST550 20 CPUs 94 GiB RAM (equivalent to 100.932 GB)
- Power Dell T430 16 CPUs 8 GB RAM
- Power Dell T330 8 CPUs 16 GB RAM

Virtual machines are KVM-based, and containers are Docker-type. The virtual machines have the same number of virtual CPUs as the physical CPUs of the corresponding host. The host and guest (virtual machines and containers) operating systems are the same, i.e., Ubuntu Server 16.04. Moreover, the power meter (hardware monitor) is the Chroma 66200.

The response time is measured on the containers, and power consumption is measured on the physical server using the hardware monitor. Since the load is equally divided among N containers, the benchmark is not considered complete until the slowest container has finished processing. Hence, the response time corresponds to the time of the slowest container, that is, the one that takes the longest to execute the portion of the load assigned to it.

To take into account the nesting of containers in virtual machines and, at the same time, distribute the load among them, we consider the different combinations between VMs and containers. For example, to distribute one third of load per container, we need to instantiate three containers that could be allocated to virtual machines as follows:

- Three containers in one virtual machine.
- One container in one virtual machine and two containers in another virtual machine.
- One container per virtual machine.

Since the CPU is the most power-demanding server component, we select CPU-intensive benchmarks, namely, sysbench, stress-ng and SPEC_CPU2017[1] to emulate the workload processed the containers [23, 24]. In detail, sysbench is a simple benchmark based on integer arithmetic that verifies prime numbers by performing standard division of the number by all numbers between two and the square root of the number itself. Stress-ng consists of a wide range of CPU specific stress tests that exercise floating point, integer, bit manipulation and control flow. In this work, we focused on the integer operations used to stress the CPU. Finally, SPEC CPU 2017 is an industry-standardized, CPU intensive benchmark consisting of four suites aimed at measuring the processor performance. In this paper, we used the floating-point-operations-based suite. This suite consists of ten real applications representative of different application domains, e.g., weather forecasting, molecular dynamics, fluid dynamics, atmosphere modeling. The applications differ in terms of their programming languages and the number of lines of their codes.

To monitor the system, we use software and hardware monitors [25]. The software monitors are used to determine the performance of the system, measured in terms of response time, while the hardware monitors measure the power consumption of the system. Then, from the response time and the power consumption, we compute the energy consumption.

4.2 Experimental Design

To determine the goodness of the nesting of virtual machines and containers, different nesting configurations are considered. Due to the physical limitations of the servers, the maximum number N of nested containers is set to six. The workload to be processed by containers is equally distributed among them. This means that in a configuration with four containers, each container will execute 1/4 of the workload. For example, in a run of the stress-ng benchmark with 100,000 stressors, each container will process 25,000 stressors.

Moreover, by considering the possible configurations of allocating containers in virtual machines, we obtained the 17 combinations (identified with a letter from A to Q) illustrated in Fig. 4. For example, combination C is the configuration corresponding to a single

[1] https://www.spec.org/cpu2017/

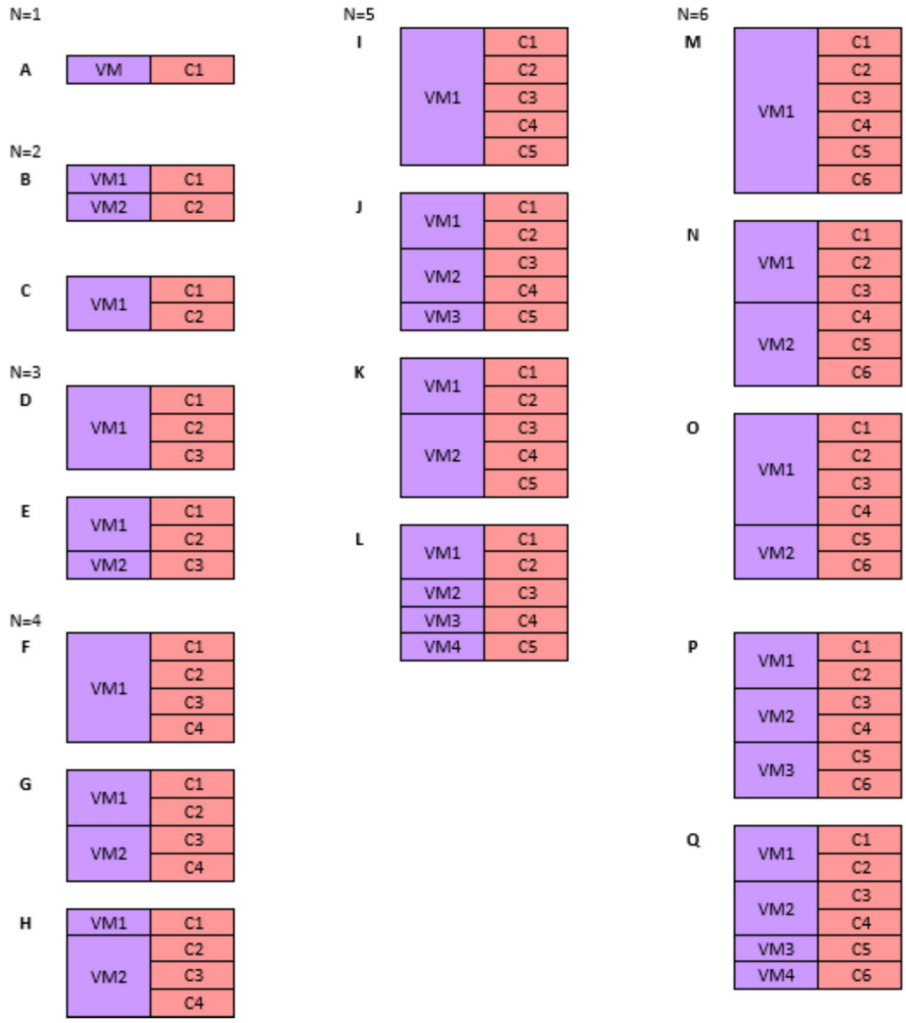

Fig. 4 Nesting configurations (i.e., 17 combinations of virtual machines and containers) considered in our experiments

virtual machine allocated in a physical machine, and two containers allocated in the virtual machine. Therefore, each container has to execute half of the workload.

5 Experimental Results

In this section, we present the results of our experiments, namely, performance, i.e, response time, as well as power and energy consumption for the three SUTs. Not to clutter the presentation, these results refer to the sysbench workload. For the other two benchmarks, we will briefly summarize the main similarities and differences. Note that to ensure the accuracy and reliability of the results, we repeat every experiment 100 times.

5.1 Performance Results

In Table 3 we summarize the response time (in seconds) as a function of the physical server and of the nesting configuration. In detail, we define the following variables (see Fig. 5 for an example):

- N: fraction of workload for each configuration.
- R_{mean} and STD: mean and standard deviation of the response time of each configuration computed over the 100 repetitions of each experiment.
- R and STD_N: mean and the standard deviation of the response time for each value of N.

Since each physical server has different hardware resources, it is interesting to analyze individual servers and understand behaviors, patterns and differences

Table 3 Summary of results obtained by the various nesting combinations as a function of the three servers considered in the experiments with the sysbench workload

		T430				T330				ST550			
	N	R	R_{meam}	STD	STD_N	R	R_{mean}	STD	STD_N	R	R_{mean}	STD	STD_N
A	1	27.709	27.709	0.000	0.000	52.675	52.675	0.000	0.000	17.054	17.054	0.000	0.000
B	2	16.817	17.188	0.063	0.523	45.893	46.839	0,077	1.337	12.190	13.104	0.066	1.293
C	2	17.558		0.057		47.784		0.380		14.018		0.031	
D	3	14.649	14.572	0.358	1.300	41.084	41.434	0.494	0.494	11.025	11.051	0.176	0.038
E	3	14.496		2.196		41.783		1.553		11.078		0.721	
F	4	12.893	11.401	0.377	0.834	31.083	29.962	0.274	1.151	8.987	8.596	0.411	0.511
G	4	10.662		1.989		28.784		0.440		8.018		0.465	
H	4	10.647		1.557		30.017		1.047		8.782		0.490	
I	5	9.673	8.900	0.388	0.574	25.978	22.931	0.912	2.034	7.345	7.628	0.162	1.149
J	5	7.083		1.713		21.978		1.026		6.257		0.358	
K	5	11.338		1.434		21.983		0.583		9.018		1.047	
L	5	7.506		1.300		21.783		0.902		7.893		1.850	
M	6	8.019	6.857	0.712	0.559	18.983	16.681	0.401	1.616	6.873	5.676	0.418	0.338
N	6	8.252		1.112		15.893		0.545		6.013		0.588	
O	6	7.566		1.873		17.673		1.275		6.458		0.960	
P	6	4.981		0.771		15.873		0.714		4.140		0.116	
Q	6	5.468		0.414		14.984		0.718		4.895		0.848	

among them. As Fig. 6 shows, the mean response time (R_{mean}) generally decreases as N increases independently of the server. This result was expected as it is mainly due to the reduction of the workload executed by the containers. Also, for a specific value of N, we have different nesting configurations of containers and virtual machines, thus the response time R varies with the configuration (see Table 3). Nevertheless, we notice that when the number of containers allocated in a virtual machine is higher than the number of containers in other configurations, the mean response time and its standard deviation tend to be bigger. In particular, for the T430 server, when $N > 2$, the variability across experiments increases. For example, for $N = 3$ (con-

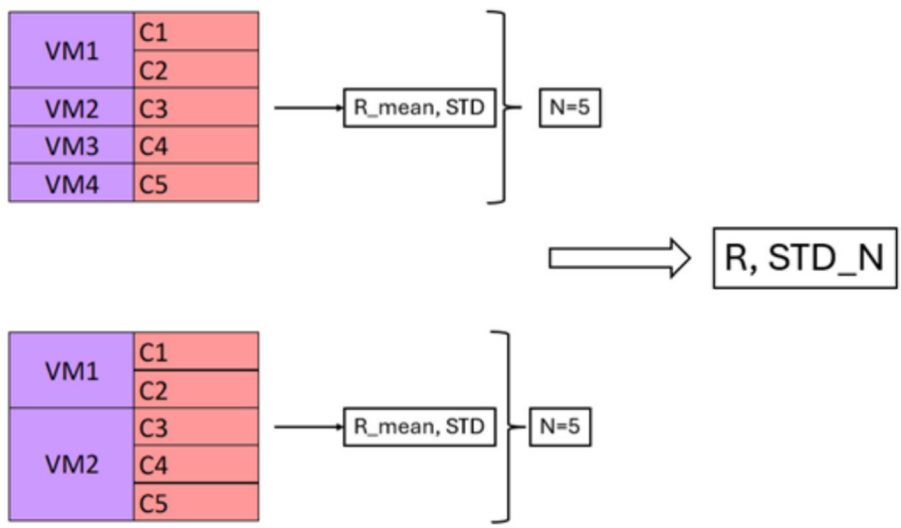

Fig. 5 Example of how R_{mean}, STD, R and STD_N are obtained for $N = 5$ with configurations K and L

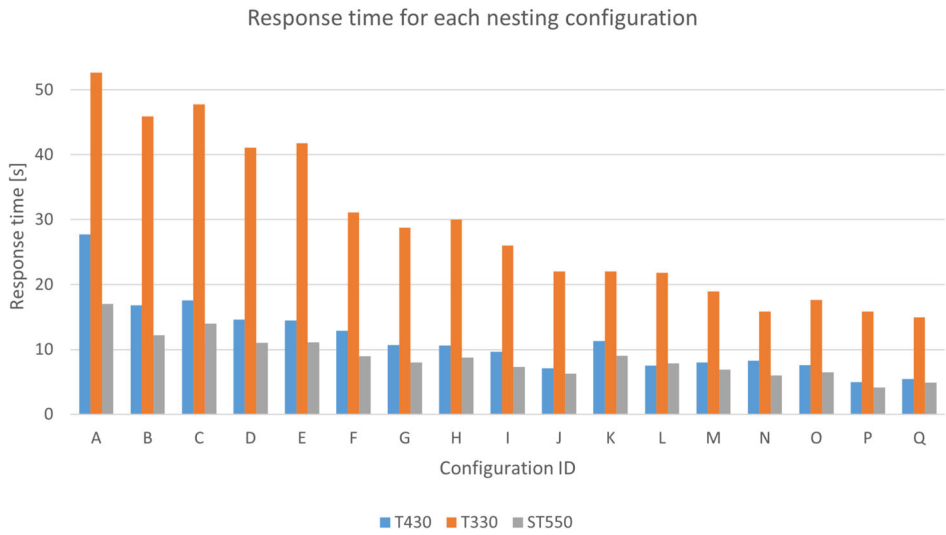

Fig. 6 Mean response time (in seconds) of the sysbench workload as a function of the physical servers and of the nesting configuration

figurations D and E), the STD_N is almost three times bigger than the standard deviation obtained for configurations B and C for $N = 2$.

Considering the value of N, for the T430 server, the most suitable option for $N = 1$ is the configuration A (one virtual machine with one container), for $N = 2$ configuration B (two virtual machines with one container each), D for $N = 3$ (one virtual machine with three containers), H or $N = 4$ (two virtual machines with one and three containers, respectively), J for $N = 5$ (one virtual machine with five containers) and P for $N = 6$ (three virtual machines with two containers each).

For the T330 server, the best configurations for nesting are: A for $N = 1$ (one container in a single virtual machine), B for $N = 2$ (two virtual machines with one container each), D for $N = 3$ (one virtual machine with three containers), H for $N = 4$ (two virtual machines with one and three containers, respectively), L for $N = 5$ (four virtual machines with two-one-one and one containers, respectively), and Q for $N = 6$ (for virtual machines with two-two-one and one containers, respectively).

For the ST550, the best configurations for nesting are: A for $N = 1$ (one container in a single virtual machine), B for $N = 2$ (two virtual machines with one container each), D for $N = 3$ (one virtual machine with three containers), G for $N = 4$ (two virtual machines with two containers each), J for $N = 5$ (three virtual machines with two, two and one containers respectively), and P for $N = 6$ (three virtual machines with two containers each).

After analyzing the different physical servers, we can conclude that under the sysbench workload the best options are the ones with fewer allocated virtual machines in a physical server, independently on the number of containers. Of course, the mean response time for ST550 server is lower than the one obtained for the T430 and T330 servers due to its more powerful hardware resources.

Regarding the other two benchmarks, stress-ng and SPEC CPU 2017, the response time behavior is very similar to what obtained for the Sysbench workload, that is, as the number of nested containers increases, the average response time decreases. In addition, this time varies for each of the nesting configurations. Configurations with a higher number of virtual machines show a higher response time due to the consolidation overhead.

5.2 Power and Energy Consumption Results

Other important attributes considered in evaluating the goodness of nesting containers in virtual machines are the energy consumption and power consumption.

It is important to recall that the power consumption is the amount of electricity that a computer hardware

draws from the power supply unit (measured in Watts). Also, the energy consumption refers to the amount of electrical energy used by a computer system to perform its operations (measured in power units · time units). This means that the energy consumption (W · s) is calculated as product of the mean power consumption (measured in Watts) and the mean response time (measured in seconds).

Table 4 presents the summary of the power and energy consumption. We outline that the mean energy consumption is calculated as the product between the mean response time and the mean power consumption. For each physical server (i.e., T430, T330 and ST550), we can easily note that the power consumption remains stable across nesting configurations. More precisely, the number of virtual machines and allocated containers do not affect the power consumption of a given server. Nevertheless, the values of the power consumption depend on the physical servers, their technology and the CPU and memory resources.

Regarding the energy consumption, the values are depicted in Fig. 7 for each server running the sysbench workload. We can see that T330 server has the highest energy consumption since this server has the worst performance in terms of response time. On the contrary, T430 and ST550 servers are more efficient in terms of energy consumption. For all servers, the average response time and energy consumption have a similar behavior. Energy consumption directly depends on response time and power consumption. As the power consumption remains constant, it does not affect the behavior of energy consumption. This means that it is the average response time that drives the behavior of the energy consumed.

5.3 Comparison Among Servers

To summarize the results presented in the previous sections, we offer in Table 5 the most suitable nesting configurations for the three servers according to the mean response time and the energy consumption. We can notice that when considering the mean response time, the physical server does not significantly affect the most suitable nesting configuration. Similarly, for energy consumption, the behavior is generally independent of the server.

Table 4 Summary of the mean response time (in seconds), power consumption (in Watts) and energy consumption (in Watts · s) of the sysbench workload for each server as a function of the nesting configuration

		T430			T330			ST550		
	N	R	Power	Energy	R	Power	Energy	R	Power	Energy
A	1	27.708	93.339	2586.321	52.674	71.054	3742.750	17.054	81.564	1391.022
B	2	16.817	94.667	1592.053	45.893	71.097	3262.921	12.189	81.894	998.250
C	2	17.557	94.074	1651.723	47.784	72.089	3444.737	14.017	81.673	1144.881
D	3	14.649	94.479	1384.032	41.084	72.684	2986.184	11.024	81.245	895.689
E	3	14.495	94.599	1371.269	41.783	73.018	3050,970	11.078	82.084	909.354
F	4	12.893	94.176	1214.249	31.083	72.674	2258,964	8.987	82.784	744.009
G	4	10.662	94.387	1006.354	28.784	72.894	2098,211	8.017	82.563	661.980
H	4	10.646	94.494	1006.030	30.017	72.674	2181.494	8.782	82.873	727.825
I	5	9.673	94.889	917.901	25.978	72.907	1894,010	7.345	82.674	607.275
J	5	7.082	94.978	672.681	21.978	72.784	1599,680	6.256	82.784	517.940
K	5	11.337	95.006	1077.131	21.983	72.984	1604.447	9.017	82.674	745.540
L	5	7.506	94.479	709.197	21.783	73.089	1592.136	7.893	82.784	653.449
M	6	8.018	94.779	760.004	18.983	73.894	1402.765	6.873	82.981	570.364
N	6	8.251	94.259	777.787	15.893	73.892	1174.400	6.013	82.784	497.815
O	6	7.565	94.023	711.368	17.673	73.672	1302.040	6.457	82.674	533.894
P	6	4.980	94.249	469.416	15.873	73.984	1174.382	4.140	83.987	347.733
Q	6	5.467	95.737	523.442	14.983	73.982	1108.515	4.895	83.235	407.470

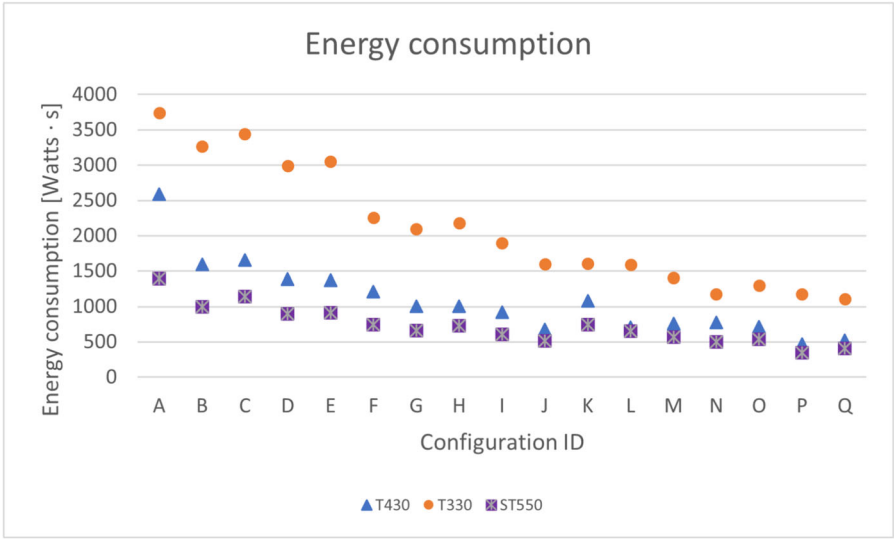

Fig. 7 Energy consumption of the sysbench workload for each server as a function of the nesting configurations

6 The Nesting Goodness Evaluation

As already stated, this paper aims to evaluate the goodness of nesting containers in virtual machines considering the trade-off between performance and energy efficiency. For this purpose, we use a metric which allows us to evaluate the goodness in terms of performance and energy consumption at the same time. In particular, to compare performance and energy consumption of N virtual machines running on a single physical machine with those of N parallel physical machines while they execute a CPU-intensive workload under saturation, we will use the CiS^2 (Consolidation Index for CPU-Server Saturation) metric introduced in Juiz and Bermejo [7]. This metric is expressed as the ratio of two EDP (Energy-Delay Product) values [26], namely:

$$CiS^2 = \frac{EDP_{VM}}{EDP_{PM}} = \frac{E_{VM} \cdot R_{VM}}{E_{PM} \cdot R_{PM}} = S_e \cdot S_p \qquad (1)$$

where EDP_{VM} and EDP_{PM} denote the Energy-Delay Products of the consolidated server and of the physical server, E_{VM} and E_{PM} represent the energy consumption, R_{VM} and R_{PM} correspond to the mean response times, and S_e and S_p denote the increment ratio of the energy and the speedup of the performance, respectively.

Let us recall that the EDP metric focuses on the overall energy consumption and performance of a system

Table 5 Comparisons between nesting configurations as a function of the server

	Response time			Energy consumption		
	T430	T330	ST550	T430	T330	ST550
N=1	A	A	A	A	A	A
N=2	B	B	B	B	B	B
N=3	D	D	D	E	D	D
N=4	H	H	G	H	G	G
N=5	J	L	J	J	J	J
N=6	P	Q	P	P	N	P

rather than on low-level metrics such as resource utilization or CPU frequency, thus it provides a high-level system-wide perspective on energy efficiency and performance. This black-box model simplifies the evaluation process and allows for a more straightforward comparison of different server options rather than getting caught up in the intricacies of individual components. Besides, the EDP metric is applicable across diverse architectures and technologies, because it allows for comparative analysis across different systems without being based on specific low-level server characteristics.

The CiS^2 index is obtained by multiplying the performance speedup (S_p) and energy ratio (S_e), thus we obtain a number that does not depend on the measurement unit and can be easily used to make comparisons [7].

In our experimental scenario, the CiS^2 index is given by the ratio between the energy of the N consolidated VMs (or containers) and the energy of N PMs multiplied by the corresponding speedup. High CiS^2 values indicate a less favorable consolidation scenario, that is, N VMs (or containers) on a single physical server are less efficient in terms of performance and energy consumption, compared to N parallel physical servers with the same workload. On the contrary, low CiS^2 values suggest that running N parallel VMs is more efficient in terms of energy and performance. By computing the CiS^2 index for different values of N, it is possible to identify the optimal point where consolidation offers the most significant benefits in terms of energy savings and performance improvement. Hence, the metric permits the evaluation of the efficiency gains achieved through virtualization and consolidation under CPU-server saturation conditions [7].

As stated in Juiz and Bermejo [7], the CiS^2 index needs a reference system to be calculated. Since we aim to evaluate the suitability of a nested system, the reference will be the system with allocated containers. It is important to note that with the CiS^2 index, a system is compared with itself, that is, the comparison of a nested server is performed with the same server without virtual machines and containers. Table 6 presents the mean response time, the power consumption and the energy consumption for each physical server as a function of the number of containers N allocated to execute the workload. The speedup calculation considers the value of N. For example, for evaluating the goodness of nesting six parallel containers ($N = 6$), the reference is the execution of the same workload into six parallel physical servers. For this reason, the power consumption is multiplied by the number of N physical servers.

6.1 Assessing the Nesting Goodness for the Sysbench Workload

Regarding the experimental results for the sysbench workload, Figure 8a plots the CiS^2 index as a function of the different nesting configurations. As can be seen, the CiS^2 values vary with the configuration because of the differences in the mean response times. We notice that the most suitable nesting configurations for the T430 server (in terms of CiS^2 value) are: A for $N = 1$, B for $N = 2$, E for $N = 3$, H for $N = 4$, J for $N = 5$, and P for $N = 6$. For the T330 server, the most suitable

Table 6 Reference system values used for the CiS^2 calculation

N	T430			T330			ST550		
	R_{mean}	Power	Energy	R_{mean}	Power	Energy	R_{mean}	Power	Energy
1	23.373	94.225	2202.340	48.543	71.202	3456.387	12.895	81.020	1044.826
2	8.858	188.868	1673.049	19.128	71.287	2727.170	4.975	80.070	796.792
3	5.029	283.320	1425.074	11.197	71.990	2418.367	2.861	82.675	709.773
4	3.369	365.342	1230.837	7.631	70.981	2166.709	1.938	79.674	617.664
5	2.476	438.545	1086.144	5.677	70.759	2008.636	1.435	79.013	567.155
6	1.934	508.254	983.319	4.473	70.463	1891.466	1.123	78.432	528.521

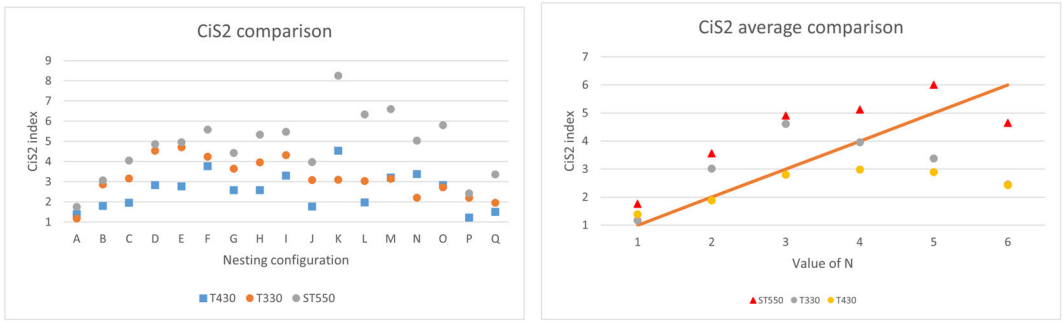

Fig. 8 CiS^2 index as a function of the nesting configuration (a) and average CiS^2 index as a function of the number of nested containers (b) for all servers

nesting configurations (in terms of CiS^2 value) are: A for $N = 1$, B for $N = 2$, D for $N = 3$, G for $N = 4$, L for $N = 5$ and Q $N = 6$. In the same way, the most suitable nesting configurations for the ST550 server (in terms of CiS^2 value) are: A for $N = 1$, B for $N = 2$, D for $N = 3$, G for $N = 4$, J for $N = 5$ and P $N = 6$.

It is important to outline that these results are similar to the results obtained by considering simultaneously the mean response time and energy consumption. Moreover, as previously stated, by means of the CiS^2 index we obtain for each physical server, the most suitable nesting configurations as a function of N. These results are very useful for system administrators who need to select a nesting configuration by considering both performance and the energy consumption at the same time.

Figure 8b shows the comparison of the mean values of CiS^2 index for the three physical servers considered in our experiments as a function of N. We can easily notice similarities and differences in the patterns. For example, for all servers, the values initially increase and then, after an inflexion point, they start decreasing. This is mainly due to the workload subdivision, that is, when N increases, the amount of workload executed by each container decreases. The value of the inflexion point is different for each physical server, that is, it occurs for a different number of nested containers (N). The workload division allows the parallelization for different values N for the three servers. For this reason, for the ST550 server, that is the most powerful machine considered in the experiments, the nesting configurations obtained for small values of N are the most inefficient because its capabilities are not fully exploited.

Moreover, we can see in Table 7 that the values of the CiS^2 index vary with N, and they are highly dependent on the physical hardware. In particular, for the server T430 for $N \geq 2$ all nesting configurations are efficient in terms of performance and energy trade-off, whereas for the T330 and ST550 servers, the efficient configurations correspond to $N \geq 4$, and $N \geq 5$, respectively. This is due to the hardware features, and especially to the RAM capacity. It would be necessary to allocate additional virtual machines with nested containers in order to obtain lower values of the CiS^2 index.

6.2 Assessment of the Nesting Goodness for the Stress-ng and SPEC CPU 2017 Workloads

To further illustrate the use of the CiS^2 metric for assessing the suitability of a nesting configuration, we considered two additional CPU workloads, namely, Stress-ng and SPEC CPU 2017. The results of the CiS^2 index for the T430 and ST550 servers and for combinations ranging from A to G (with N up to 4), are presented in Table 8.

These results confirm the results obtained for the sysbench benchmark (see Section 6.1). In particular, the values of the CiS^2 index change with the nesting combinations and with the physical servers. For the SPEC CPU 2017 workload the most suitable nesting combinations are A, B, D and F, for N equal to 1, 2, 3 and 4, respectively. For the Stress-ng workload with

Table 7 CiS^2 index for the three servers for the sysbench workload

	N	T430		T330		ST550	
		CiS^2	CiS^2_{avg}	CiS^2	CiS^2_{avg}	CiS^2	CiS^2_{avg}
A	1	1.392	1.392	1.175	1.175	1.760	1.760
B	2	1.806	1.881	2.870	3.013	3.069	3.558
C	2	1.956		3.155		4.048	
D	3	2.828	2.800	4.530	4.618	4.861	4.910
E	3	2.773		4.707		4.959	
F	4	3,775	2.981	4.246	3.953	5.585	5.119
G	4	2.587		3.652		4.433	
H	4	2.582		3.960		5.339	
I	5	3.300	2.897	4.314	3.382	5.478	6.012
J	5	1.771		3.083		3.979	
K	5	4.539		3.092		8.257	
L	5	1.978		3.041		6.334	
M	6	3.203	2.427	3.146	2.447	6.604	4.648
N	6	3.373		2.205		5.043	
O	6	2.829		2.719		5.808	
P	6	1.228		2.202		2.425	
Q	6	1.504		1.962		3.360	

80K operations, the most suitable nesting combinations are A, C, E and G for N equal to 1, 2, 3 and 4, respectively, while A, B, D and F for the workload with 160K operations.

Moreover, we can observe that the inflexion point has the same behavior of the sysbench workload. For the physical servers with more powerful hardware, it would be necessary to allocate additional virtual machines with additional nested containers to investigate the down effect after the inflexion point.

7 Discussion

In this section, we summarize and discuss the most relevant conclusions of our experimental study. First of all, the experiments suggest that the trend of the performance is independent of the physical servers characteristics and the workload being executed, that is, when N increases, the mean response time tends to decrease mainly because of the workload division. In addition, depending on the nesting configuration, the mean response time varies from server to server. Nev-

Table 8 CiS^2 index for SPEC CPU 2017 and Stress-ng workloads

	CiS^2 index		
	T430	ST550	
Nesting combination	SPEC_CPU	Stress-ng(80K)	Stress-ng(160K)
A	1.55	1.03	1.01
B	3.04	1.41	1.95
C	6.04	1.18	2.62
D	2.91	1.79	3.72
E	4.35	0,90	4.07
F	2.43	0,21	4.54
G	2.83	0.18	5.11

ertheless, as a general behavior, the fewer the allocated virtual machines in a physical server, the better the performance. Moreover, as expected, the mean response time depends on the physical server characteristics, namely, the higher the number of CPUs and the bigger the RAM capacity, the lower the mean response time.

Hence, from our experiments, we can conclude that to select a nesting configuration considering the mean response time, it is crucial to have a similar number of virtual machines and containers, independently of the physical server resources and of the workloads.

The second relevant result refers to the power consumption. Our experiments have shown that for each physical server the power consumption does not vary for the various nesting configurations. We can indeed observe a trend, that is, when the number of containers increases, the power consumption tends to grow to a limited extent. Therefore, the power consumption does not play a significant role in the selection of a nesting configuration.

The third important result refers to the energy consumption. As previously mentioned, the energy consumption is proportional to the mean response time. Since the power consumption remains constant, the energy consumption and the mean response time have a similar behavior. As a consequence, to select a nesting configuration that takes into account the energy consumption, it is necessary to consider first the mean response time.

Other interesting conclusions refer to the goodness of nesting configurations. For a given value of N, the values of the CiS^2 index vary as a function of the nesting configuration, thus the most suitable configuration is the one with the lowest CiS^2 index. In addition, the values of the CiS^2 index vary with N because of the workload division. This behavior is independent of the physical server and the workload. Therefore, the CiS^2 index can be used to determine the goodness of nesting containers in virtual machines for every hardware and workload types.

As a general rule, whatever the way to consolidate is (virtual machines, containers or nesting virtual machines and containers) the overhead should be minimum. Hence, the number of virtual machines and containers should be as low as possible. In the case of nesting, the number of virtual machines and containers should be as balanced as possible and the value of the CiS^2 index should be minimum. Then, we can state that the behavior of the goodness of server consolidation considering simultaneously the performance and the energy consumption does not vary among the different ways to consolidate, as [7, 11] demonstrated previously. This means that the CiS^2 index behavior is independent of how the consolidation is made, using virtual machines, containers or nesting containers in virtual machines.

This study has some potential limitations. For example, the use of KVM and Docker, as virtualization technologies, might have affected performance, energy consumption and the CiS^2 index. Moreover, experiments performed on real infrastructures are often affected by the issues inherent to real experimentation, such as sampling performance and power consumption. However, despite these limitations, previous works (e.g., [7, 11]) demonstrated that the consolidation overhead behavior is independent of hardware, workload type and hypervisor.

8 Conclusion and Future Work

This work studied the goodness of server consolidation in terms of performance and energy consumption considering the current trend of nesting containers in virtual machines while processing CPU-intensive workloads. We investigated the suitability of these solutions for the current consolidated and virtualized servers. To achieve this aim, real experimentation was performed using classical benchmarking and monitoring techniques.

To investigate the energy consumption, three CPU-intensive workload were considered (i.e., Sysbench, Stress-ng and SPEC CPU 2017). In addition, we defined a set of configurations varying the number of virtual machines and containers together with nesting combinations. From these configurations, the performance (mean response time) and the power consumption were measured through software and hardware monitors. Moreover, we applied the CiS^2 index to quantify the trade-off between performance and energy consumption and the suitability of each nesting configuration.

From the experimental results, we noticed similar behaviors in the mean response time, power consumption, and energy consumption for all nesting combinations. However, the values of the CiS^2 index indicated that the suitability of the various configurations varies

and depends on the number of virtual machines and nested containers. In general, to obtain a good trade-off it is advisable to use similar numbers of virtual machines and containers.

The results obtained in this work will be very useful for system's administrators who will be able to decide about virtual machines and containers nesting without having information about their datacenters. Hence, we believe that the use of the proposed approach will have a potential influence on datacenter management.

As future research activities, we plan to extend this work by increasing the number of virtual machines and containers using a simulation approach. Besides, we will extend this work by varying the virtualization technology and the characteristics of the workloads being executed. In fact, the proposed methodology is not suitable for I/O, memory or communication bound workloads because for these workload types, the CPU utilization is usually very low and the power consumption of the physical servers remains at its nominal value, thus there are no effects on the energy consumption and the CiS2 metric is meaningless.

Acknowledgements This work is part of Project TED2021-132695B-I00, funded by MCIN / AEI / 10.13039 / 501100011033 and by the European Union "NextGenerationEU" / PRTR.

Author contributions These authors contributed equally to this work.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work is part of Project TED2021-132695B-I00, funded by MCIN / AEI / 10.13039 / 501100011033 and by the European Union "NextGenerationEU" / PRTR.

Data Availability No datasets were generated or analysed during the current study.

Code availability Not applicable

Declarations

Some journals require declarations to be submitted in a standardised format. Please check the Instructions for Authors of the journal to which you are submitting to see if you need to complete this section. If yes, your manuscript must contain the following sections under the heading 'Declarations':

Competing Interests The authors declare ni competing interests.

Ethics Approval and Consent to Participate Not applicable

Consent for Publication Not applicable

Materials Availability Not applicable

References

1. Buyya, R., Broberg, J., Goscinski, A.M. (eds.): Cloud Computing: Principles and Paradigms. Wiley Series on Parallel and Distributed Computing. Wiley, Hoboken, New Jersey (2011)
2. Juiz, C., Capo, B., Bermejo, B., Fernández-Montes, A., Fernández-Cerero, D.: A case study of transactional workload running in virtual machines: the performance evaluation of a flight seats availability service. IEEE Access. **11**, 81600–81612 (2023)
3. Higgins, J., Holmes, V., Venters, C.: Securing user defined containers for scientific computing. In: International Conference on High Performance Computing & Simulation (HPCS), pp. 449–453 (2016). IEEE
4. Combe, T., Martin, A., Di Pietro, R.: To Docker or not to Docker: A Security Perspective. IEEE Cloud Computing. **3**(5), 54–62 (2016)
5. Bermejo, B., Juiz, C.: On the classification and quantification of server consolidation overheads. J. Supercomput. **77**(1), 23–43 (2021)
6. Bermejo, B., Juiz, C.: Virtual machine consolidation: a systematic review of its overhead influencing factors. J. Supercomput. **76**(1), 324–361 (2020)
7. Juiz, C., Bermejo, B.: The CiS^2: a new metric for performance and energy trade-off in consolidated servers. Clust. Comput. **23**(4), 2769–2788 (2020)
8. Randal, A.: The ideal versus the real: Revisiting the history of virtual machines and containers. ACM Computing Surveys. **53**(1) (2020)
9. Pautasso, C., Zimmermann, O., Amundsen, M., Lewis, J., Josuttis, N.: Microservices in practice, part 1: Reality check and service design. IEEE Softw. **34**(1), 91–98 (2017). https://doi.org/10.1109/MS.2017.24
10. Raza, S.M., Jeong, J., Kim, M., Kang, B., Choo, H.: Empirical performance and energy consumption evaluation of container solutions on resource constrained iot gateways. Sensors. **21**(4), 1378 (2021)
11. Bermejo, B., Juiz, C.: A general method for evaluating the overhead when consolidating servers: performance degradation in virtual machines and containers. J. Supercomput. **78**(9), 11345–11372 (2022)
12. Shah, S.A.R., Waqas, A., Kim, M.-H., Kim, T.-H., Yoon, H., Noh, S.-Y.: Benchmarking and performance evaluations on various configurations of virtual machine and containers

for cloud-based scientific workloads. Appl. Sci. **11**(3), 993 (2021)
13. Xavier, M.G., Neves, M.V., Rossi, F.D., Ferreto, T.C., Lange, T., De Rose, C.A.: Performance evaluation of container-based virtualization for high performance computing environments. In: 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 233–240 (2013). IEEE
14. Seo, K.-T., Hwang, H.-S., Moon, I.-Y., Kwon, O.-Y., Kim, B.-J.: Performance comparison analysis of Linux container and virtual machine for building cloud. Advanced Science and Technology Letters. **66**, 105–111 (2014)
15. Li, Z., Kihl, M., Lu, Q., Andersson, J.A.: Performance overhead comparison between hypervisor and container based virtualization. In: 31st International Conference on Advanced Information Networking and Applications (AINA), pp. 955–962 (2017). IEEE
16. Ruan, B., Huang, H., Wu, S., Jin, H.: A performance study of containers in cloud environment. In: Wang, G., Han, Y., Martínez Pérez, G. (eds.) Advances in Services Computing, pp. 343–356. Springer, Cham (2016)
17. Mavridis, I., Karatza, H.: Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing. Futur. Gener. Comput. Syst. **94**, 674–696 (2019)
18. Shah, S.A.R., Waqas, A., Kim, M.-H., Kim, T.-H., Yoon, H., Noh, S.-Y.: Benchmarking and performance evaluations on various configurations of virtual machine and containers for cloud-based scientific workloads. Appl. Sci. **11**(3), 993 (2021)
19. Barik, R.K., Lenka, R.K., Rao, K.R., Ghose, D.: Performance analysis of virtual machines and containers in cloud computing. In: International Conference on Computing, Communication and Automation (ICCCA), pp. 1204–1210 (2016). IEEE
20. Mavridis, I., Karatza, H.: Performance and overhead study of containers running on top of virtual machines. In: 19th Conference on Business Informatics (CBI), vol. 2, pp. 32–38 (2017). IEEE
21. Cuadrado-Cordero, I., Orgerie, A.-C., Menaud, J.-M.: Comparative experimental analysis of the quality-of-service and energy-efficiency of vms and containers' consolidation for cloud applications. In: 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–6 (2017). IEEE
22. Jain, R.: The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley, New York (1991)
23. Casalicchio, E.: A study on performance measures for auto-scaling CPU-intensive containerized applications. Clust. Comput. **22**(3), 995–1006 (2019)
24. Bucek, J., Lange, K.-D., Kistowski, J.: SPEC CPU2017: Next-generation compute benchmark. In: Companion of the ACM/SPEC International Conference on Performance Engineering, pp. 41–42 (2018)
25. Calzarossa, M.C., Massari, L., Tessera, D.: Performance monitoring guidelines. In: Companion of the ACM/SPEC International Conference on Performance Engineering, pp. 109–114 (2021)
26. Juiz, C., Bermejo, B.: On the scalability of the speedup considering the overhead of consolidating virtual machines in servers for data centers. The Journal of Supercomputing. (2024)