The Journal of Supercomputing (2022) 78:11345–11372
https://doi.org/10.1007/s11227-022-04318-5

A general method for evaluating the overhead when consolidating servers: performance degradation in virtual machines and containers

Belen Bermejo[1] · **Carlos Juiz**[1]

Accepted: 12 January 2022 / Published online: 8 February 2022

Abstract

Server consolidation is one of the most commonly used techniques for reducing energy consumption in datacenters; however, this results in inherent performance degradation due to the coallocation of virtual servers, i.e., virtual machines (VMs) and containers, in physical ones. Given the widespread use of containers and their combination with VMs, it is necessary to quantify the performance degradation in these new consolidation scenarios, as this information will help system administrators make decisions based on server performance management. In this paper, a general method for quantifying performance degradation, that is, server overhead, is proposed for arbitrary consolidation scenarios. To demonstrate the applicability of the method, we develop a set of experiments with varying combinations of VMs, containers, and workload demands. From the results, we can obtain a suitable method for quantifying performance degradation that can be implemented as a recursive algorithm. From the set of experiments addressing the hypothetical consolidation scenarios, we show that the overhead depends not only on the type of hypervisor and the workload distribution but also on the combination of VMs and containers and their nesting, if feasible.

Keywords Overhead · Server consolidation · Virtualization · Performance evaluation

✉ Belen Bermejo
belen.bermejo@uib.es

Carlos Juiz
cjuiz@uib.es

[1] Computer Science Department, University of the Balearic Islands, Palma de Mallorca, Spain

1 Introduction

Due to the increase in Internet service use, datacenters and servers need to be managed more efficiently. Such management should consider not only performance but also energy efficiency; since 20% of IT company budgets cover electricity waste, power and energy consumption are currently the main concerns of datacenter administrators. Moreover, global IT CO2 emissions represent 7% of all worldwide emissions [8].

To mitigate the impact of IT energy consumption, green IT has been conceptualized as a set of techniques for using IT in a greener manner; one such example is server consolidation, in which the maximum workload is allocated to the minimum number of physical servers. Virtualization technology drives server consolidation using virtual machines (VMs) or containers. Despite the similar functionality between VMs and containers, there are significant differences between them in terms of performance (measured by the mean response time in this work), security, deployment, and portability. These differences affect consolidation decisions when choosing between VMs and containers and the consolidation number that should be implemented.

Traditionally, servers are consolidated by allocating either VMs or containers to a physical server. However, the combination of VMs and containers in the same physical server can mitigate the drawbacks of both. Accordingly, it is possible to first consolidate containers into VMs and then consolidate these VMs in physical machines [9].

Many authors have demonstrated that although server consolidation has positive effects on power consumption and energy efficiency, it incurs performance drawbacks [12] 15 10. Specifically, server consolidation results in inherent performance degradation due to the extra software layers generating during the virtualization (the hypervisor and consolidation), together with the resource's usage. The magnitude of this performance degradation depends on the consolidation approach; that is, whether VMs, containers, or some combination of the two is implemented. Every consolidation combination has its particularities (functional and nonfunctional requirements) and, in turn, its own level of performance degradation due to consolidation overhead. In [2], the authors demonstrated the importance of considering the consolidation overhead due to its large impact on response time [14]. In addition, they proposed the first approach to quantifying the consolidation overhead only in VM-alone and container-alone scenarios without considering heterogeneous combinations of VMs and containers or the allocation of containers into VMs.

Considering previous works, this study aims to quantify the server consolidation overhead for any arbitrary consolidation combination of VMs and containers. That is, we propose a general method for quantifying and graphically representing the consolidation overhead from the perspective of the physical server. Since every consolidation combination (called configuration) has individual requirements, we also experiment with several combinations (and nesting) of VMs and containers to determine the most suitable combination (in terms of overhead) for deployment in a specific consolidation scenario. Accordingly, the research questions we attempt to answer are as follows:

- *RQ1* Is there a general method for quantifying the server consolidation overhead regardless of its configuration?
- *RQ2* For a particular nesting level of consolidation, which is the amount of overhead?

To answer the above questions, we contribute a general method for quantifying the server consolidation overhead. In addition, we separately explore the behavior and values of the overhead for virtualization (OV_v) and consolidation (OV_c).

In a previous work [2], the authors proposed a method for quantifying the consolidation overhead. However, this method cannot be applied to any arbitrary consolidation configuration since it is limited to either VM consolidation or container consolidation. Therefore, in this work, we extend the previous method to any kind of consolidation, including combinations of different numbers of VMs and containers.

This text is organized as follows. In Sect. 1, we introduced the context of the problem and the research questions. In Sect. 2, we review the related work to clarify the problem to be addressed. Then, in Sect. 3, we present the main concepts needed to understand this work. The problem statement is explained in Sect. 4, followed by the overhead quantification method and its algorithmic implementation in Sects. 5 and 6, respectively. In Sect. 7, we evaluate the proposed method and discuss the main findings. Then, we discuss the previous results in Sect. 8. Finally, we offer conclusions and discuss future work in Sect. 9.

2 Related work

Several studies have been conducted to address the effect of consolidation overhead on performance degradation in consolidated systems and the quality of service and to compare the performance of VMs and containers, as the latter need a light software layer to be deployed. In [4], the authors classified existing research studies on performance comparisons between VM hypervisors and containers. However, these performance comparisons were made from the perspective of application without considering the set of software layers supported by the physical server.

Similarly, in [6], the authors compared the performance of KVM and Docker and concluded that Docker offers better performance from the application perspective. In addition, in [13], the feasibility of containers in high-performance applications was explored by comparing the performance of commonly used container technologies such as LXC and Docker.

The above works considered performance from the perspective of the application executed in a VM or a container. From this perspective, the performance of containers is better than that of VMs. In contrast, in our work, we consider the performance from the perspective of the physical server, which contains and supports any virtualization platforms in datacenters.

In terms of physical server performance degradation, in [12], the authors investigated the disk I/O performance and its isolation by comparing VM deployment and container deployment and primarily found that VMs outperform containers in terms

of I/O performance and isolation in a DBMS. This finding is contrary to the general belief that containers perform better than VMs because of their lack of virtualization overhead.

As seen in previous works, the server consolidation affects the performance degradation. In [10], the authors studied the most influential factors affecting server consolidation. In addition, in [15], the authors reviewed the state-of-the-art research on managing the performance overhead of VMs to reveal the causes of VM performance overhead. Considering these factors, in [2], the authors classified the types of consolidation overhead (overhead of virtualization and overhead of consolidation) and proposed a general method for estimating their values. This classification method can be applied to systems with a variety of server characteristics and workload types; however, it can only be applied to consolidations based on either VMs or containers; combinations cannot be considered.

To the best of our knowledge, this work is the first attempt to propose a general method for quantifying and representing the consolidation overhead for any consolidation configuration, that is, VMs, containers or containers inside VMs.

3 Background

In this section, we review the concepts needed to understand this research work. First, we review server consolidation, followed by the virtualization technology that enables server consolidation. Finally, we review the concept of consolidation overhead as the primary subject of this work.

3.1 Server consolidation

Server consolidation is a technique that helps system administrators flexibly manage servers and datacenters by attempting to allocate the maximum workload (VMs or containers) to the minimum number of physical servers. Traditionally, servers are consolidated via VM consolidation, in which VMs are reallocated among different physical machines (PMs). For example, in Fig. 1, we present three physical servers, all of which have VMs allocated to them. In this case, VMs from server A and server C can be migrated to server B (which has sufficient resources to support them). In this way, the utilization of physical resources in server B increases, while servers A and C can be switched off, reducing the corresponding power consumption [3] 9.

3.2 Virtual machine- and container-based virtualization

At the infrastructure level, VMs serve as the backbone of the cloud. A virtual machine can be defined as the simulation of a computer device created by emulation software to execute an application. VM-based virtualization is the most commonly used technique in the cloud environment, in which physical resources are virtually distributed at the hardware level through a hypervisor [3].

A hypervisor or virtual machine monitor (VMM) allows multiple operating systems to share a single PM through one or more VMs. The management layer controls all instantiated VMs, each of which runs an independent operating system. There are two main types of hypervisors: Type I (see Fig. 2a) and Type II (see Fig. 2b). A Type I (or bare metal) hypervisor is software running directly on a hardware platform, while a Type II (or hosted) hypervisor is software that runs inside another operating system [3].

A Type I hypervisor runs directly on the hardware of the PM and does not require a host OS to run; consequently, it requires additional resources to operate. By avoiding an extra software layer between the host hardware and the VM, a Type I hypervisor performs better than a Type II hypervisor. Additionally, a Type I hypervisor is more secure and resource efficient than a Type II hypervisor [12]. Another recent form of virtualization is based on containers. Containerization is a method for virtualizing an OS without a hypervisor (see Fig. 3b). Virtual instances of an OS, called containers, are a form of isolating the OS environment and its file system that runs on a single host and a single kernel.

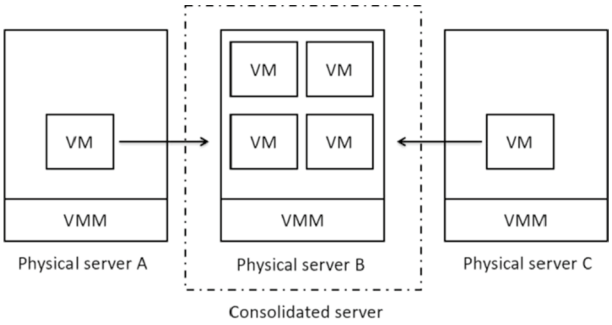

Fig. 1 Traditional server consolidation

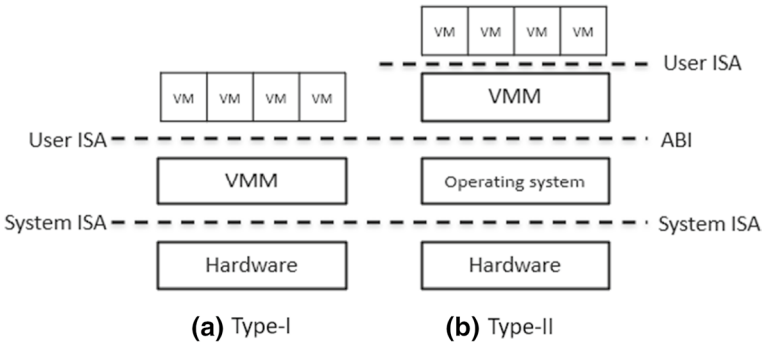

Fig. 2 Type I and Type II hypervisors (based on [3])

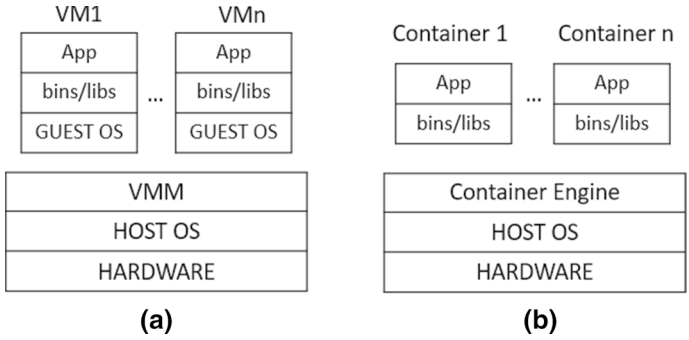

Fig. 3 **a** Hypervisor-based virtualization vs. **b** Container-based virtualization

3.3 Consolidation overhead

The consolidation overhead is the extra workload that the system incurs for managing several VMs or containers (current access to the server's resources).

In [2], the authors stated that there are various factors influencing the server consolidation overhead, such as the hypervisor type and the features of the VM.

In summary, regarding the performance degradation of VMs and containers,

[9] reports that containers are superior to VMs in terms of performance, scalability, energy consumption, and provider profit. However, containerization suffers from weak isolation, which may create significant security problems. These issues can be solved by running containers on top of VMs. While VMs offer strong isolation, the main advantage of containers is the low overhead achieved, as they share the same operating system kernel, which increases the percentage of consolidation overhead. Nevertheless, the performance overhead cost is still important in both VM architectures and architectures involving some combination of VMs and containers [2].

4 Problem statement

Since server consolidation results in inherent overhead, in [2], the authors proposed a general method for estimating the values of the different types of consolidation overhead regardless of the virtualization platform, server characteristics, or workload type.

The server consolidation overhead is defined as the extra workload that the system incurs to support consolidation, regardless of the hypervisor type (based on VMs or containers). There are two types of overhead: OV_v is the overhead inherent to the virtualization technology (hypervisor), and OV_c is the overhead resulting from the set of coallocated virtual instances (VMs or containers).

At present, virtual servers can be deployed in the form of either VMs or containers on a physical server. Hence, consolidation in cloud datacenters consists of grouping objects, such as VMs, containers, or data, to occupy unused resources.

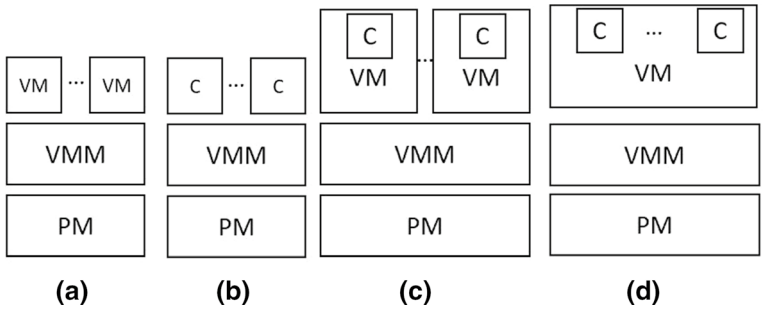

Fig. 4 Different forms of consolidation

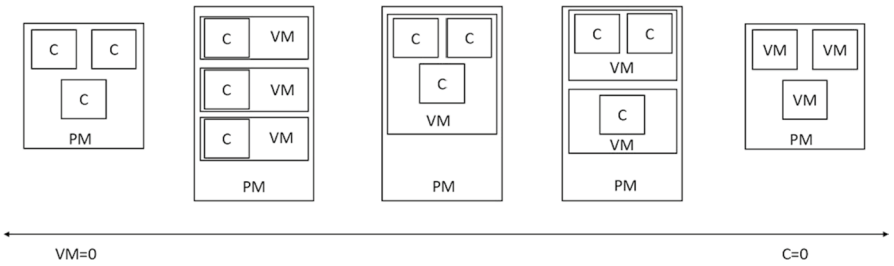

Fig. 5 Combinations for three virtual servers (N = 3)

According to the literature [9], there are several possible consolidation combinations. It is important to note that containers can be allocated within a VM, but not vice versa. In this work, we are interested in the possible infrastructure consolidation combinations, which can be characterized as follows:

- *VM-PM* Consolidation of VMs within PMs (see Fig. 4a).
- *Container-PM* Consolidation of multiple containers within a set of PMs (see Fig. 4b).
- *Container-VM* Consolidation of multiple containers within a set of VMs (see Fig. 4c).
- *Container-VM and VM-PM* Consolidation of containers within VMs and those VMs within PMs (see Fig. 4d).

The combinations represented in Fig. 4 can be extended to consider different numbers of VMs and containers. For example, in Fig. 5, two different consolidation scenarios are depicted for three virtual servers. On the leftmost side, we consider a scenario in which three containers are consolidated within a single physical machine (from Fig. 4b). On the rightmost side, we consider a scenario in which three VMs are consolidated within a single physical machine (from Fig. 4a). Between these two scenarios, we can consider combinations of different numbers of VMs and containers: three VMs with a single container each, one VM with three containers, and two VMs, one with two containers and the other with a single container.

The example presented in Fig. 5 can be extended to any degree of consolidation. As many authors have stated [7], 1, VMs and containers have different purposes and features.

Therefore, is there a general method for quantifying server consolidation overhead regardless of the configuration? Moreover, for a particular nesting level of consolidation, what is the amount of overhead?

5 Overhead quantification method

As stated previously, to the best of our knowledge, there is a need for a general method for quantifying the consolidation overhead for any consolidation scenario. In [2], the authors proposed a method for quantifying server consolidation overhead for consolidation with only VMs or containers (see Fig. 6). Any other possible consolidation scenarios were not considered in the pro- posed method, limiting the knowledge about the corresponding overhead.

Under the previously introduced model, the two types of overheads (OV_v and OV_c) are calculated as shown below (see Eqs. 1 and 2), where R^P is the mean response time of the workload when executed on the physical server, R^V is the mean response time of the workload when executed in an isolated VM or container, and R^C is the mean response time in the consolidation scenario. It is important to note that R^V arises from an unrealistic scenario because there is no practical reason to deploy a physical server with a single VM or container. Nevertheless, in the method proposed in this work, this scenario does have practical meaning since it represents any consolidation scenario.

$$\mathrm{OV}_v = R^V - R^P \tag{1}$$

$$\mathrm{OV}_c = R^C - R^V \tag{2}$$

As seen above, the two overheads are calculated as the difference between the mean response times in the different virtualization scenarios representing an evolution in virtualization deployment (from a physical server to a consolidated server): (1) any kind of virtualization, (2) a single VM or container, and (3) the consolidation scenario of interest. For instance, we can generalize this

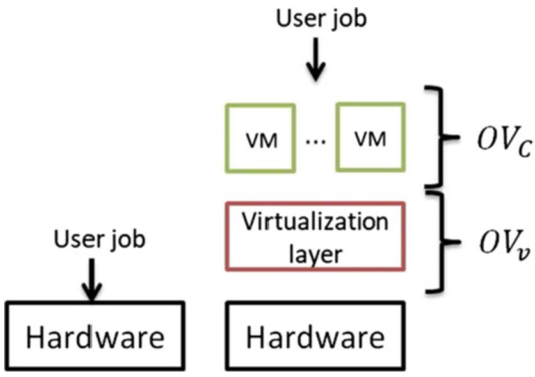

Fig. 6 Performance overhead due to consolidation

virtualization deployment evolution to consider any consolidation scenario described in the previous section. Every step in this evolution consists of the addition of a virtualization or consolidation layer.

For example, in Fig. 7, an evolution of consolidation scenarios is depicted. Scenario 1 depicts only a physical machine; this scenario has no virtualization, which means no possibility of consolidation. Then, on this hardware, a hypervisor and a single VM are deployed (scenario 2). A single container can then be allocated to this single VM (scenario 3). Finally, a set of containers can be consolidated within this VM (scenario 4). In this case, scenarios 2 and 3 are needed to calculate the consolidation overhead of scenario 4. In the same manner, to determine the consolidation overhead of scenario 3, scenarios 1 and 2 are needed.

In summary, to determine the overhead of a consolidation configuration, we need the two previous levels that lead to the final configuration. Let us take R^C as the mean response time of the consolidation configuration, R^B as the mean response time of the lower of the two previous levels (the base case), and R^V as the mean response time of the intermediate level. In the previous example, R^B corresponds to scenario 2, R^V corresponds to scenario 3, and R^C corresponds to scenario 4. Therefore, the relationship among the different layers is as follows. The mean response time in the consolidated scenario is the result of adding the overhead of virtualization and the overhead of consolidation to that of the base case. In addition, the mean response time in the base case is the difference between the mean response time in the intermediate scenario and the virtualization overhead (see Eqs. 3 and 4).

$$R^B = R^V - \mathrm{OV}_v \tag{3}$$

$$R^C = \left(\mathrm{OV}_v + \mathrm{OV}_c\right) + R^B \tag{4}$$

From Eq. 3, we can derive R^V: $R^v = R^B + \mathrm{OV}_v$. Therefore, the virtualization overhead and consolidation overhead can be defined as Eqs. 5 and 6, respectively, allowing us to quantify the different types of consolidation overhead regardless of the consolidation scenario.

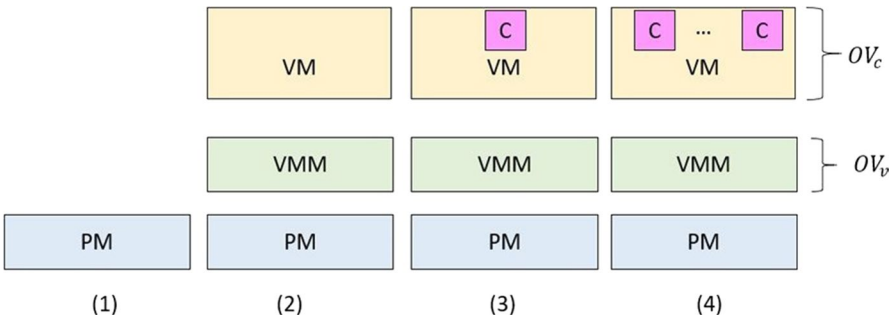

Fig. 7 Consolidation scenario evolution

$$\mathrm{OV}_v = R^V - R^B \tag{5}$$

$$\mathrm{OV}_c = R^C - R^V \tag{6}$$

As the previous formulation shows, the overhead calculation can be automated for a specific consolidation scenario when the two previous levels of virtualization are known. In the next section, we propose an algorithm for implementing this calculation.

6 Overhead determination algorithm

Since the two previous levels are needed to calculate the consolidation overhead of a specific consolidation configuration, a recursive algorithm can be defined to perform this calculation. The algorithm requires the consolidation configuration and its mean response time (R^C); if the two previous levels have not yet been explored, the algorithm will recursively work backward (among the different layers) to calculate the mean response time values to finally obtain OV_v and OV_c.

In the proposed algorithm, we assume that two levels lower than the target consolidation configuration exist in order to calculate OV_v and OV_c, as these levels are necessary to calculate the virtualization and consolidation overheads. However, to increase the robustness of the algorithm without changing its nature, we have added three default exceptions: when there is no server (layer 0), when there is a physical server but no virtualization (layer 1), and when there is virtualization but no consolidation (layer 2). As long as at least three layers exist, the proposed method can be applied for any type of virtualization and consolidation. Considering the previous exceptions, the whole algorithm can be depicted as shown in Algorithm 1.

Algorithm 1 *Response time (R layer, initial layer)- Generic*

```
Require: double: R layer, integer: initial layer
  integer OVc=0
  integer OVv=0
  integer layer=0
  layer = initial layer
  if layer >= 0 then
    if initial layer==0 then
      print "there is no server"
    else if initial layer==1 then
      print "there is no virtualization"
    else if initial layer==2 then
      print "there is no consolidation"
      OV v = responsetime((Rlayer), layer) —responsetime((Rlayer - 1), layer - 1)
    else if initiallayer >= 3 then
      OV c = Rlayer —responsetime((Rlayer - 1), layer - 1)
      OV v = responsetime((Rlayer —1), layer -1)- responsetime((Rlayer -2), layer -2)
    end if
  end if
  return OVv
  return OVc
```

6.1 Experimental setup

To demonstrate the proposed algorithm, we use a set of values obtained from real experiments.

The experimental setup is composed of two types of physical servers using the Intel Xeon E5-2600 CPU family: (1) a Dell PowerEdge T430 (or a set of these physical servers), with 16 physical CPUs, 8 GB of RAM, and Ubuntu Server 16.04 as the operating system, and (2) a Dell PowerEdge T330 (or a set of these physical servers), with 8 physical CPUs, 16 GB of RAM, and Ubuntu Server 16.04 as the operating system.

For virtualization, we deploy KVM as a Type I hypervisor, Virtual Box as a Type II hypervisor, and Docker as a container-based hypervisor. All VMs and containers are assigned the same number of CPUs as the physical server, 1 GB of virtual RAM, and Ubuntu Server 16.04 as the guest operating system. The two executed workloads are from the Sysbench and Stress-ng benchmarks, both of which are CPU intensive [5]. It is important to note that in this work, the executed workload requests 100% CPU utilization, representing CPU sat uration.

6.2 Application of the method and algorithm (examples)

In the next sections, we illustrate the use of the proposed algorithm by considering two scenarios: one with the same and the other with different numbers of VMs and containers. It is important to note that in Figs. 8 and 9, the 'V' path refers to the next-lowest level of virtualization and the 'B' path is the lowest level.

6.2.1 Case 1: identical numbers of VMs and containers

As an example of the application of the previously introduced algorithm, the third combination approach for consolidation depicted in Fig. 7 is used in this section (see Fig. 8a). This combination consists of a physical machine on which three VMs are allocated, with a single container for each of them.

As stated previously, to determine OV_v and OV_c for a specific consolidation case, the two previous virtualization levels are needed. For this example, these levels are shown in Fig. 8b and c. The intermediate level is composed of the PM and three consolidated VMs, while the bottom level is composed of the PM with a single allocated VM.

Utilizing values obtained from real experiments (with KVM and Docker), the mean response times for each level are $R^C = 14,\ 339\ s$, $R^V = 13,\ 838\ s$ and $R^B = 5,\ 889\ s$. By applying the previous formulation and considering the response time of the target consolidation configuration, the values of OV_v and OV_c are obtained as follows:

$$\mathrm{OV}_v = 13,838 - 5,889 = 7,949s \rightarrow 55,43\%$$
$$\mathrm{OV}_c = 14,330 - 13,838 = 0,492s \rightarrow 3,433\%$$

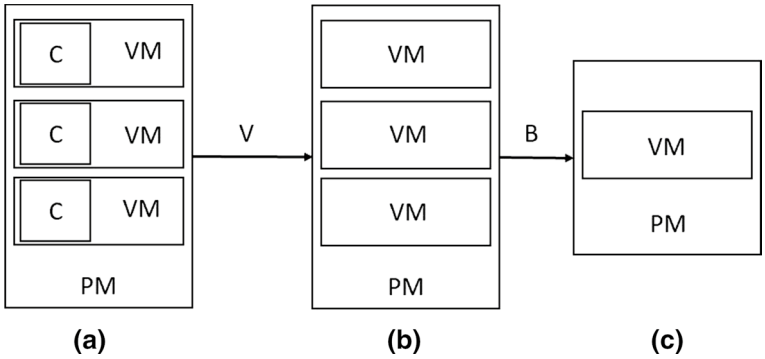

Fig. 8 Case 1: representation of three virtual servers, with one container per VM

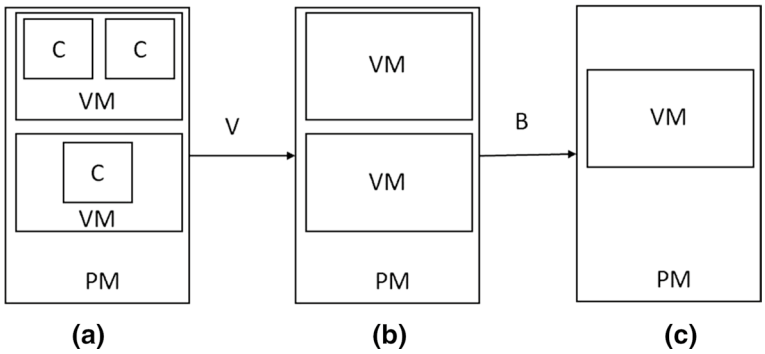

Fig. 9 Case 2: representation of three virtual servers, with two containers in one VM and the remaining container in another

The obtained values mean that for the selected consolidation configuration, 55,43% of the time, the PM is devoted to managing the virtualization platform, that is, managing access to the VMs from the containers and access to the PM from the VMs. Moreover, the physical machine uses 3,433% of its time to manage the consolidated containers in the VMs. Consequently, the PM uses only 41,05% of its time to execute the workload.

Since less than 1/2 of the available time is devoted to workload execution and the rest of the time is devoted to consolidation and virtualization management, it is crucial to consider the corresponding overhead. Moreover, from the previous example, we can see that the overhead from the VMs is higher than that from the containers.

6.2.2 Case 2: different numbers of VMs and containers

In contrast to the previous case, we consider the consolidation configuration represented in Fig. 9a, which contains two VMs: one with one container and the other with two containers. In the same manner as in the previous subsection, the two

required levels for OV_v and OV_c determination are represented in Fig. 9b and c (the intermediate and lowest levels, respectively).

Utilizing values obtained from real experiments (with KVM and Docker), the mean response times for each level are $R^C = 39,\ 602\ s$, $R^V = 38,\ 577\ s$ and $R^B = 25,\ 426\ s$. By applying the previous formulation and considering the response time of the target consolidation configuration, the values of OV_v and OV_c are obtained as follows:

$$\mathrm{OV}_v = 38,577 - 25,426 = 13,151s \rightarrow 34,09\%$$
$$\mathrm{OV}_c = 39,602 - 38,577 = 1,025s \rightarrow 2,58\%$$

In this case, 34,09% of the time is devoted to managing the virtualization platform, and 2,58% of the time is needed to support the consolidation of the containers. Therefore, only 63,33% of the time is used to execute the workload.

7 Evaluation method

In this section, we evaluate the proposed method of quantifying the consolidation overheads (OV_v and OV_c). As mentioned previously, this method can be applied to any consolidation configuration involving some combination of VMs and containers using Type I, Type II, or container-based virtualization on any physical machine executing workloads of any nature, amount, and distribution.

To this end, we conducted a set of real experiments exploring a wide range of scenarios involving variations in the workload distribution, the hypervisor type, and the consolidation combination. The workload distribution may be either proportional or nonproportional. That is, every machine (physical, virtual, or container) will execute an n-th part of the workload (uniformly), but each part may have a different nature and intensity. For example, if we have to perform 100 mathematical operations, each PM (or VM or container) will execute 25 operations. In the case of a proportional distribution, each operation will be of the same nature (for example, addition). However, if the distribution is nonproportional, one operation could be addition, another could be a trigonometrical operation, and so on.

The workload is distributed uniformly among the different VMs and containers since they have the same importance within the server. However, even if the same amount of workload is executed within each virtual server, the intensity of the workload may not be the same. This situation reflects the distribution of users within a datacenter. Each server may serve the same number of users, but their tasks may have different intensities. The behavior of users with different intensities is reflected by the Sysbench benchmark (nonproportional distribution), whereas users with the same intensity are reflected by the Stress-ng benchmark (proportional distribution), varying the *–cpu-ops* parameter. For the scenarios detailed below, we calculated the OV_v and OV_c values from the measured mean response times using a software monitor in the physical server:

1. Proportional workload distribution

2. Nonproportional workload distribution
    a. Equal numbers of VMs and containers
    b. Unequal numbers of VMs and containers

Throughout this section, the values of OV_v and OV_c (overhead metrics) are reported together with the useful work time, that is, the amount of the time during which the system is actually executing the workload (in this case, the CPU operations).

7.1 Results for proportional workload distributions

For proportional workload distributions, the server consolidation configurations are evaluated for a set of scenarios (see Fig. 10) under the Stress-ng workload execution. These scenarios involve a variety of virtual servers, as follows:

- *Scenario 1* A set of N homogeneous physical servers.
- *Scenario 2* A set of N homogeneous VMs consolidated on a single physical server.
- *Scenario 3* A set of N homogeneous containers consolidated on a single physical server.

In every scenario, the single workload is distributed proportionally among the different physical machines, VMs, or containers. That is, every physical server (or VM or container) executes the same workload (same number of CPU operations) at the same intensity (same kind of operation). The system capacity is distributed based on the processor sharing discipline [11] (the difference being that there is only a single client or workload). Specifically, in scenario 1, the workload is executed in parallel among the N PM, and a similar uniform distribution is implemented among the virtual servers in scenarios 2 and 3. To obtain a wide spectrum of results, we performed experiments using two different physical servers, a Dell PowerEdge T430 and a Dell PowerEdge T330, whose features were explained in the previous section. In addition, for each server, we used Type I, Type II, and container-based hypervisors.

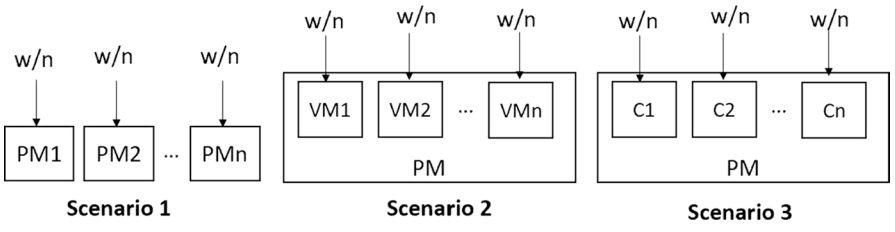

Fig. 10 Evaluation scenarios for a proportional workload distribution

7.1.1 Results for the T330 server

The results presented in this subsection correspond to the Dell PowerEdge T330 server. In Fig. 11, the overhead metrics for the Type I hypervisor are graphically represented as percentages as a function of the number of consolidated VMs. We can observe that OV_v does not have a significant impact on the global overhead, regardless of the number of VMs. In contrast, the value of OV_c increases as the number of consolidated VMs grows. This behavior is due to the increase in the number of VMs that must be managed. Consequently, the percentage of time that can be devoted to useful work decreases as the number of VMs increases. If a greater proportion of the mean response time is devoted to managing the consolidation, a lesser proportion of time can be devoted to executing the workload.

In the same manner, in Fig. 12, the overhead metrics are represented for the Type II hypervisor. As in the Type I case, OV_v does not have a significant impact on the total consolidation overhead. However, the value of OV_c initially increases with the number of consolidated machines, reaching a maximum when the number of consolidated machines is three (N = 3). As N continues to increase, the percentage of time available for useful work remains constant for different numbers of VMs.

Regarding container consolidation, we can observe in Fig. 13 that the consolidation overheads (OV_v and OV_c) are not significant. That is, the mean response time is almost entirely used for workload execution. The behavior explained previously is illustrated in detail by presenting the absolute magnitudes (in seconds) in Table 1.

It is important to note that OV_v, OV_c and the useful work percentages from the previous figures are normalized values. Although some overhead values are very low when expressed as percentages, the absolute values are not (see Table 1). In this case, the Type II hypervisor yields the highest value for OV_v, regardless of the number of VMs. This is due to the hypervisor implementation, which requires an OS between the VMs and the hardware. Regarding OV_c, the Type II hypervisor also results in higher values. However, container consolidation provides a higher percentage of useful work time.

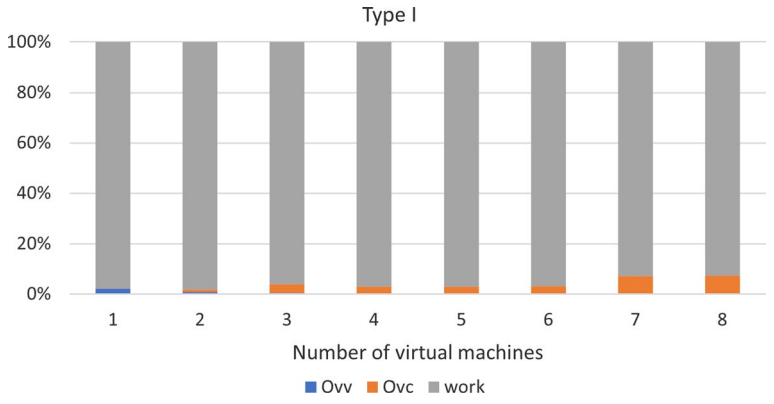

Fig. 11 Overhead metrics for a Type I hypervisor on the T330 server

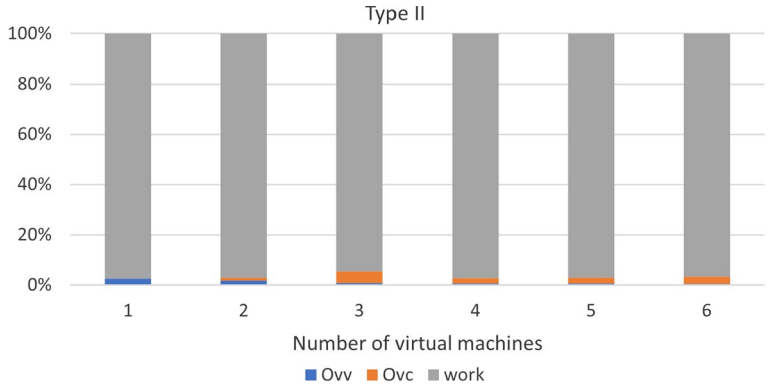

Fig. 12 Overhead metrics for a Type II hypervisor on the T330 server

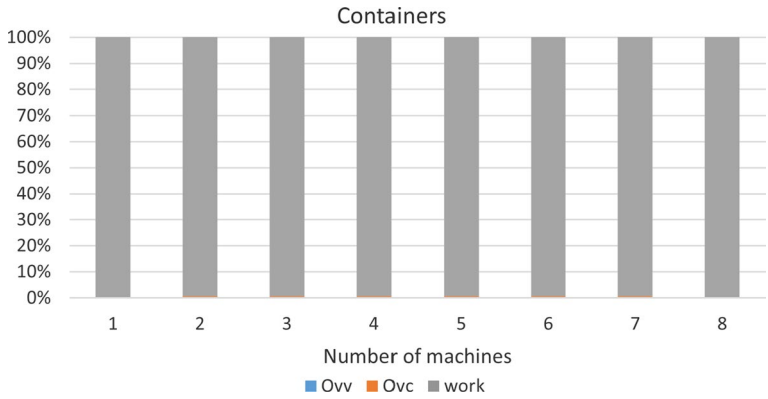

Fig. 13 Overhead metrics for containers on the T330 server

Another factor to consider is the consolidation degree of the different hypervisors. The Type I and container-based hypervisors can support up to eight consolidated VMs. However, the Type II hypervisor can consolidate only six VMs. This limitation is due to the virtualization implementation, as detailed previously.

7.1.2 Results for the T430 server

The results presented in this subsection correspond to the Dell PowerEdge T430 server. In Fig. 14, the overhead metrics are graphically represented for the Type I hypervisor as a function of the number of consolidated VMs. In this case, we can observe that the value of OV_v is not significant, being less than 1% regardless of the number of consolidated VMs. Similarly, the value of OV_c does not exceed 2% for any degree of consolidation. Additionally, both values remain stable as the number of VMs grows. In consequence, most of the mean response time is devoted to

Table 1 Overhead metrics for Type I, Type II, and container-based hypervisors on the T330 server (in seconds)

	Type I			Type II			Containers		
N	OVv	OVc	Work	OVv	OVc	Work	OVv	OVc	Work
1	17.99	0.00	811.69	21.28	0.00	811.69	0.45	0.00	811.69
2	7.47	7.81	822.21	14.81	10.26	818.16	0.23	4.97	811.91
3	3.05	30.60	826.63	7.76	39.55	825.21	0.66	3.58	811.48
4	2.04	23.86	827.64	5.59	18.50	827.38	0.38	4.61	811.76
5	1.62	23.66	828.06	4.77	20.66	828.20	1.20	4.38	810.94
6	1.40	25.22	828.28	3.72	24.00	829.25	1.00	3.24	811.14
7	1.32	63.54	828.36	3.48			0.77	4.20	811.37
8	1.16	65.31	828.52	3.11			0.12	1.04	812.02

executing the workload. The percentage of useful work time increases smoothly with an increase in the number of consolidated VMs.

In the same manner, in Fig. 15, the overhead metrics for the Type II hypervisor are depicted as a function of the number of consolidated VMs. We can observe that the value of OV_v decreases smoothly as the number of VMs grows. Moreover, OV_c behaves in the same manner. Therefore, the percentage of useful work time increases with the number of VMs.

In Fig. 16, the overhead metrics are represented for container consolidation. We observe that the value of OV_v is higher than it is for Type I or Type II consolidations and decreases with an increasing number of virtual servers. The value of OV_c also decreases with an increasing number of containers but is always less than OV_v. Consequently, the percentage of useful work time increases as the number of containers increases. As in the case of the Type I hypervisor, this behavior is due to the distribution (proportional) of the workload among the containers.

In Table 2, the absolute values of the overhead metrics are shown. We observe that the values of OV_v for container consolidation are higher than those for Type I and Type II consolidation. This demonstrates that the high number of software layers needed for container deployment affects performance. In addition, the mean response time for container consolidation is higher than the Type I and Type II mean response times.

As in the case of the T330 server, the consolidation degree varies among the hypervisor types, with the Type II hypervisor being less capable of consolidating a large number of VMs. Moreover, the performance of the T330 server is lower than that of the T430 server, having a higher mean response time. This is due to the number of physical CPUs in the T430 server, which is higher than that in the T330 server.

7.2 Results for nonproportional workload distributions

For nonproportional workload distributions, server consolidation configurations are evaluated by varying the number of consolidated virtual servers for different numbers of VMs and containers under the Sysbench-CPU workload execution as follows (see Fig. 17):

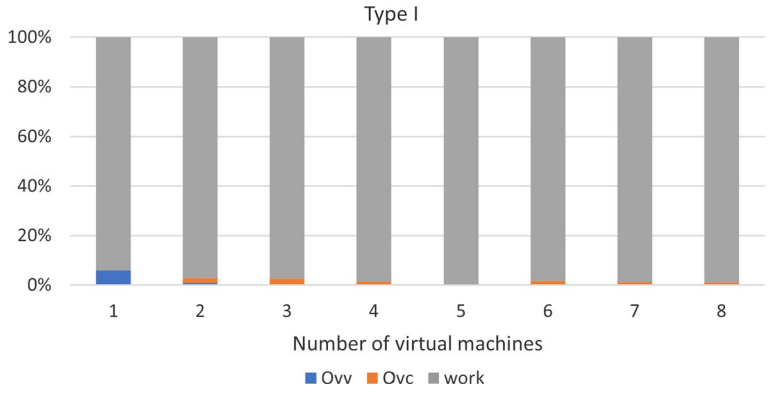

Fig. 14 Overhead metrics for the Type I hypervisor on the T430 server

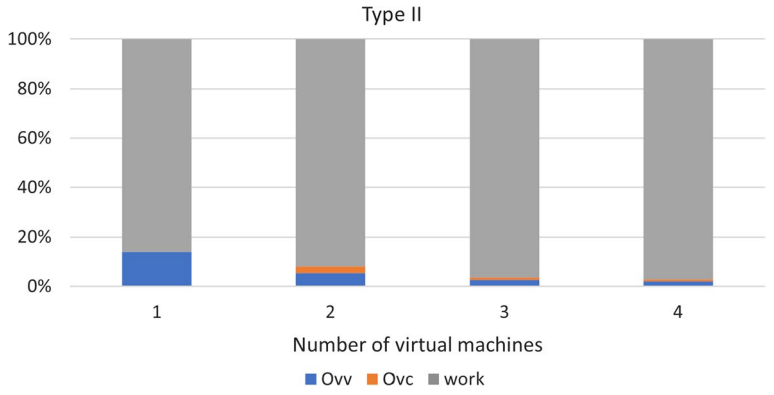

Fig. 15 Overhead metrics for the Type II hypervisor on the T430 server

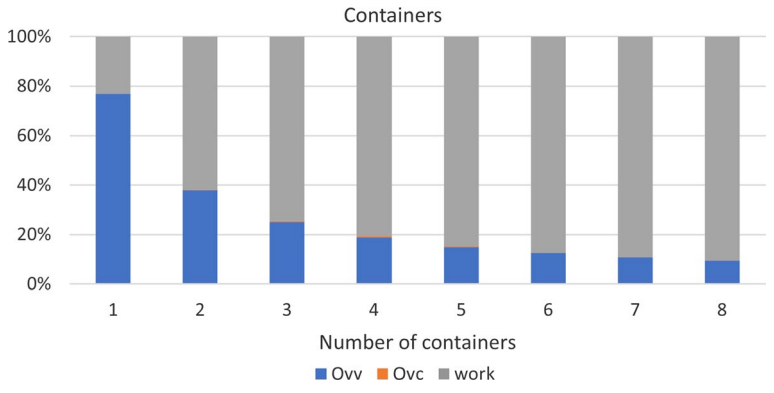

Fig. 16 Overhead metrics for containers on the T430 server

- *Scenario 1* A set of N homogeneous physical servers (Dell Power Edge T430).
- *Scenario 2* A set of N homogeneous VMs consolidated within a single physical server. In this scenario, the hypervisors used are Type I (KVM) and Type II (Virtual Box).
- *Scenario 3* A set of N homogeneous containers consolidated within a single physical server. In this scenario, a container-based hypervisor (Docker) is used.
- *Scenario 4* A set of N homogeneous containers consolidated within M homogeneous VMs, in turn consolidated within a single physical server. In this scenario, the VMs are virtualized with KVM and the containers are virtualized with Docker.

In all scenarios, the single workload is distributed in a balanced manner among the different physical machines, VMs, or containers. That is, the distributed parts of the workload are the same size, but the intensity differs. Since in this case we are considering the Sysbench workload, some of these parts could have more complex prime-number operations than others.

The system capacity is distributed based on the processor sharing discipline [11] (with the difference being that there is only a single client or workload). Specifically, in scenario 1, the workload is executed in parallel among the N physical machines, and it is similarly distributed among the virtual servers in scenarios 2 and 3. In scenario 4, there is a combination of M VMs and N containers, and the workload is proportionally divided among the latter.

7.2.1 Equal numbers of VMs and containers

In this subsection, the results for equal numbers of VMs and containers are presented. The different depicted metrics were measured in scenarios 1, 2, 3, and 4. For scenario 4, two combinations of VMs and containers are represented by two subscenarios: scenario 4.1 includes 1 PM, 1 VM, and N containers (PM/VM-container), whereas scenario 4.2 includes 1 PM and N VMs, each with one container (PM-(VM/container)).

Table 2 Overhead metrics for Type I, Type II, and container-based hypervisors on the T430 server (in seconds)

	Type I			Type II			Containers		
N	OVv	OVc	work	OVv	OVc	work	OVv	OVc	work
1	6.91	0.00	110.06	17.84	0.00	110.06	368.99	0.00	110.06
2	1.03	2.44	115.93	6.98	3.66	120.92	182.54	1.22	296.51
3	0.20	3.10	116.76	3.36	1.34	124.53	121.06	1.31	357.98
4	0.33	1.44	116.64	2.62	1.18	125.27	90.74	3.35	388.30
5	0.08	0.15	116.88	1.59			72.12	1.83	406.92
6	0.19	1.74	116.78	1.85			60.64	0.24	418.40
7	0.17	1.34	116.80	1.33			51.76	0.63	427.28
8	0.30	0.94	116.67	1.32			45.37	0.26	433.67

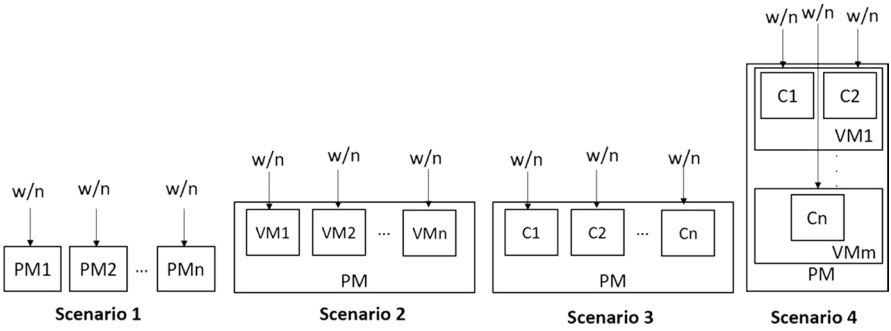

Fig. 17 Consolidation scenarios for experimentation

In Fig. 18, the overhead metrics are represented as a function of the number of consolidated VMs. We observe that the OV_v value does not have a significant impact on the total overhead. However, the value of OV_c depends on the number of consolidated VMs. From N = 2 to N = 5, OV_c increases and then decreases from N = 6 to N = 9. This is due to the balance between the workload division (nonproportional) and the number of consolidated machines. Consequently, the percentage of useful work time decreases from N = 2 to N = 5 and then varies from N = 6 to N = 9.

In the same manner, in Fig. 19, the overhead metrics are represented as a function of the number of VMs. As with the Type I hypervisor, OV_v does not have a significant impact on the total overhead. However, OV_c represents more than 33% of the mean response time. In this case, the value of OV_c increases as the number of VMs grows until N = 5. As the number of consolidated VMs increases, more accesses to resources must be managed. Then, at N = 6, OV_c decreases due to the balance between the workload division and the number of consolidated machines. As a consequence, the percentage of useful work time decreases until N = 5 and then increases again.

Regarding container consolidation, in Fig. 20, we present the overhead metrics as a function of the number of consolidated containers. In this case, the value of OV_v reaches 15% when N = 2 and then decreases to less than 4%. The OV_c value oscillates between 12 and 33% for $3 <= N <= 8$. Then, for N = 9, the OV_c value is minimal. Therefore, the value of the useful work percentage ranges from 83 to 99%, depending on the number of consolidated containers and the nonproportional workload features.

As in the case with a proportional workload distribution, we observe that Type I and container consolidation reach a higher degree of consolidation than Type II consolidation up to N = 9 and N = 6, respectively. This is due mainly to the hypervisor implementation, but the nonproportional workload distribution also plays a role. The nonproportional distribution implies that not all parts of the workload have the same features, with some being more demanding than others in terms of resources.

Previously, we stated that the allocation of containers to VMs has advantages in terms of functionality. Now, we consider the two combinations introduced above: scenario 4.1 (PM/VM-container) and scenario 4.2 (PM-(VM/container)). In Fig. 21,

the overhead metrics are represented for the PM/VM-container case. In this case, the value of OV_v remains between 1 and 1.8% from N = 2 to N = 9. The corresponding impact of OV_v on the total consolidation overhead is very low. However, for $N > 2$, the value of OV_c is between 34 and 57%. As a consequence, the percentage of useful work time oscillates between 50 and 63% for $N > 2$.

In this case, the overhead metric values depend on the number of consolidated containers in the single VM and the intensity of the different parts of the workload. In addition, it is important to note that the OV_c value is higher than in Type I, Type II, or container consolidation due to the allocation of the containers to a single VM. This finding indicates that the improvement in functionality is adversely affected by that allocation.

In the same manner, in Fig. 22, we present the overhead metrics for PM- (VM/container) consolidation, where we vary the number of VM-container blocks within the PM. In this case, the value of OV_v ranges from 1 to 1.3% for N = 2 to N = 5 and does not have a significant impact on the total consolidation overhead. On the other hand, the value of OV_c ranges from 22 to 45% for N = 2 to N = 5 and increases with the number of consolidation blocks. As a consequence, the percentage of useful work time decreases with an increasing number of consolidation blocks.

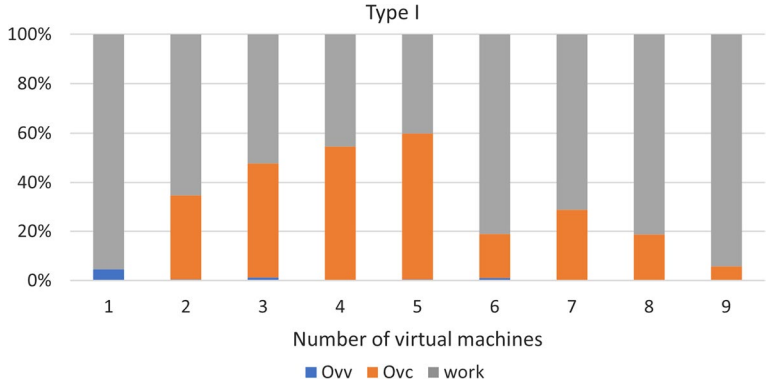

Fig. 18 Overhead metrics for the Type I hypervisor

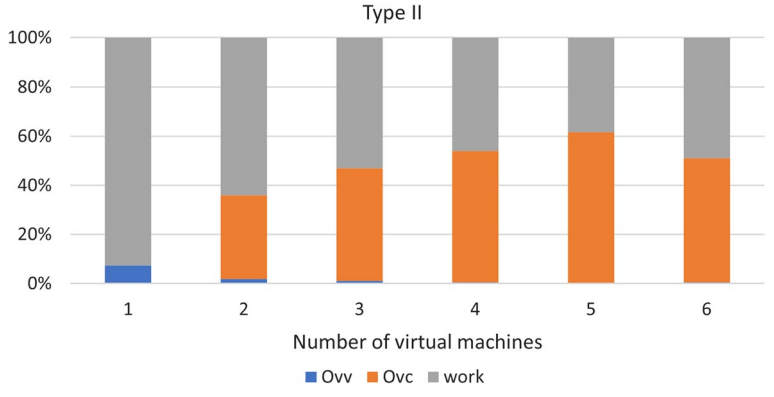

Fig. 19 Overhead metrics for the Type II hypervisor

Moreover, we observe that in this case, only up to five consolidated VMs can be established, whereas in the previous case (PM/VM container), we can reach up to 9 consolidated containers. This is due to the lightweight nature of the containers' implementation.

We summarize the results presented in this section in Table 3, where the percentage of useful work time is listed for each hypervisor type and each consolidation degree. Additionally, we indicate the most efficient hypervisor type for each consolidation degree. The greater the percentage of useful work time, the more efficient the configuration is. In this case, container consolidation results in less consolidation overhead than the other configurations.

It is important to note that a higher percentage of useful work time alone does not necessarily imply better performance. In this case, container consolidation is efficient from the perspective of the useful work percentage, but it has worse performance than the Type I hypervisor, for example. In Table 4, we list the mean response time (absolute) of the consolidation configuration for each hypervisor type. From this, we can observe that although container consolidation offers a higher percentage of useful work time, it also results in a higher mean response time.

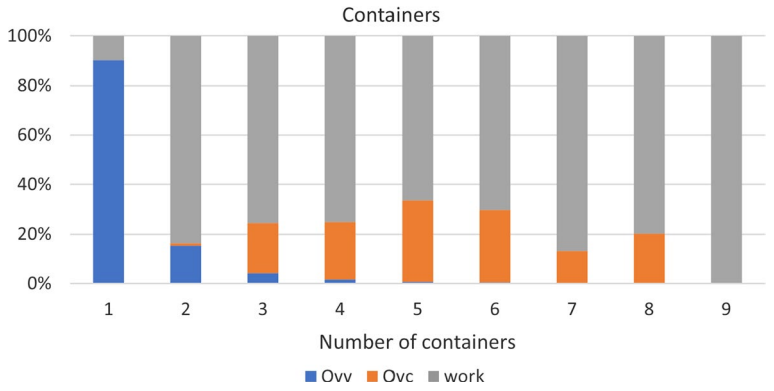

Fig. 20 Overhead metrics for containers

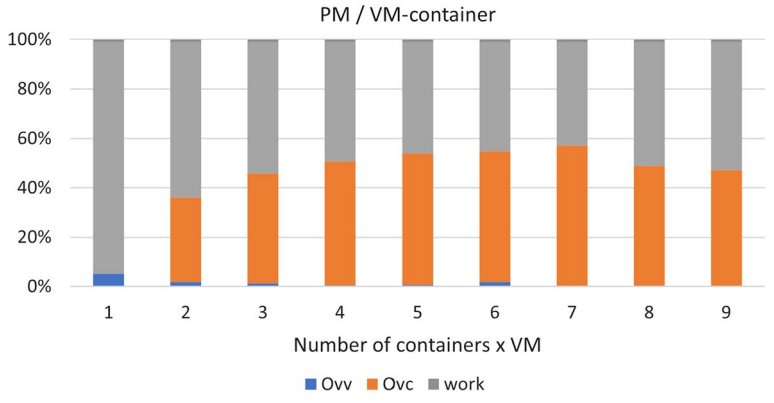

Fig. 21 Overhead metrics for PM/(VM-container) consolidation

7.2.2 Unequal numbers of VMs and containers

In this case study, we study different consolidation scenarios considering different combinations of VMs and containers (see Fig. 23). All these VMs and containers are allocated to a single physical machine. The features of the physical server, VMs, and containers are the same as in the first case study.

For this case study, we select the case of N = 6; that is, we wish to consolidate 6 execution blocks using VMs and containers. We vary the degree of consolidation from N = 1 to N = 6 to build different configurations as follows. For example, in case K, VM1 and VM2 are allocated to a single physical machine, C1 and C2 are allocated to VM1, and C3, C4, and C5 are allocated to VM2. In any configuration, the workload is divided in a balanced manner among the containers for execution, as in the previous case study. For example, in case K, each container executes 1/5th of the whole workload.

In this case, OV_v and OV_c are represented in Fig. 24. Moreover, the time dedicated to useful work is also represented (work). The overhead and useful work metrics are all represented as normalized variables (between 0 and 1).

We observe in Fig. 24 that for any consolidation configuration, OVv does not have a significant impact on the total overhead. However, OV_c makes a large contribution to the consolidation overhead. Scenarios D, F, I, K, M, N, and P have the highest OV_c percentages. This is due to the degree of container consolidation for each VM, which is higher in those scenarios than in the others. That is, the greater the number of containers allocated to a VM, the more hardware resources (and time) are needed to manage resource access for the containers. As a result, scenarios A, B, C, E, G, J, O, and Q allow higher percentages of useful work time and are more suitable for workload consolidation than the other scenarios. Also, the scenarios H, L, and R present very similar behavior among them. The percentage of OV_c is higher than the percentage of useful work.

Moreover, in Table 5, the mean response time for each consolidation scenario is listed. We observe that the mean response time decreases as the number of

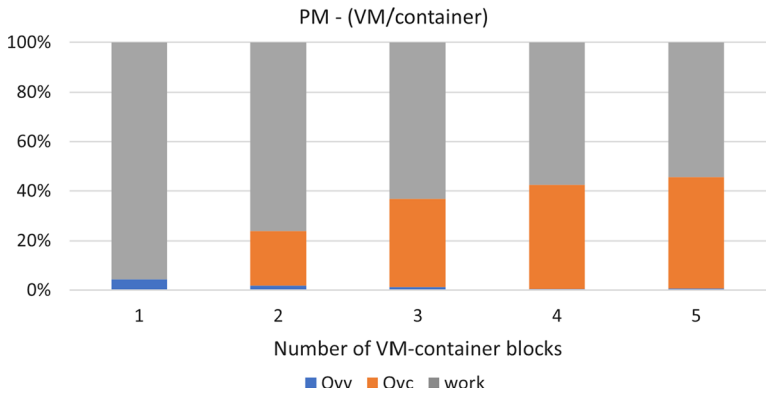

Fig. 22 Overhead metrics for PM-(VM/container) consolidation

Table 3 Percentage of useful work time by scenarios

N	Hypervisor type				
	Type I	Type II	Containers	PM/(VM-container)	PM-(VM/container)
1	0.95	0.92	0.90	0.94	0.95
2	0.65	0.64	0.83	0.63	0.76
3	0.52	0.52	0.75	0.54	0.63
4	0.45	0.45	0.75	0.49	0.57
5	0.39	0.38	0.66	0.45	0.54
6	0.80	0.48	0.70	0.45	
7	0.71		0.86	0.42	
8	0.81		0.79	0.50	
9	0.94		0.99	0.52	

Table 4 Mean response time of the different scenarios

N	Hypervisor type				
	Type I	Type II	Containers	PM/(VM-container)	PM-(VM/container)
1	25.42	26.21	247.62	23.48	27.70
2	19.28	19.83	125.07	17.86	17.77
3	15.83	16.12	103.74	14.10	14.33
4	13.97	14.06	80.62	11.78	11.95
5	12.54	13.58	73.93	10.08	10.08
6	5.15	8.83	58.50	8.31	
7	5.07		40.62	7.81	
8	3.89		38.75	5.73	
9	2.99		27.51	4.92	

containers increases due to the workload division. In addition, a balanced distribution of containers among the VMs corresponds to a shorter mean response time.

8 Discussion

In previous sections, the experimentation results were analyzed, yielding a number of conclusions regarding consolidation overhead.

To apply the proposed method, a set of real measurements is needed. The real measurements should be taken from three different layers, represented by three different consolidation scenarios. If a real system is not available, necessary for monitoring its activity running the benchmarks, the overhead quantification could have limited results.

OV_v includes the task related to the hypervisor instantiation, which depends on the type (VM-based on containers-based). Also, in OV_c, all the contention regarding the applications (or workload) executed in the VMs or containers are included. Since the proposed method attempts to quantify the consolidation overhead, isolation of the different contention effects is not necessary.

Regarding the consolidation values, we observed that among the different experiments, they depend on the number of consolidated VMs, containers and their

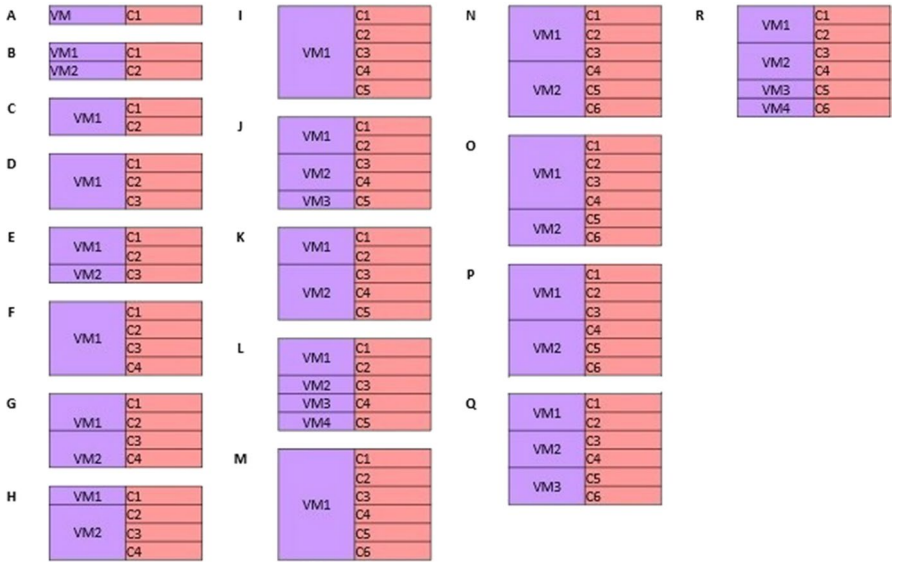

Fig. 23 Combinations of VMs and containers

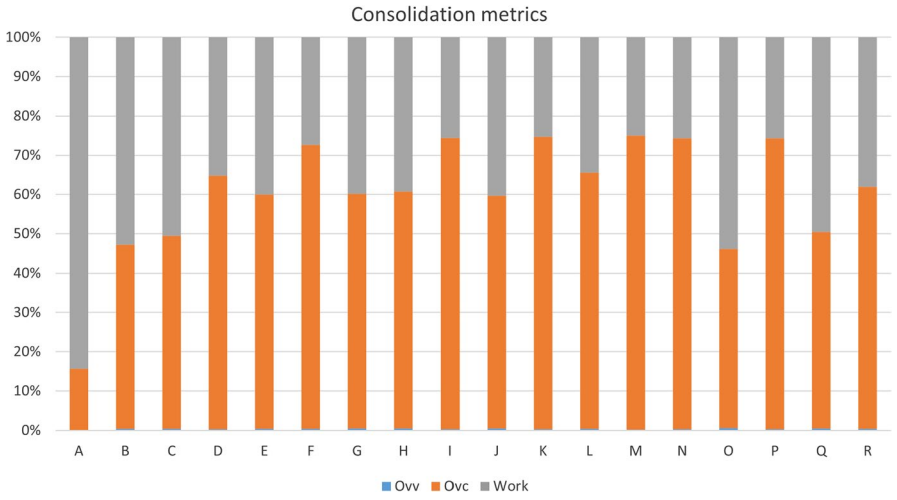

Fig. 24 Percentage overhead and useful work metrics

combinations and on the physical server features. In Sect. 6.1, we detailed the physical server features; the T430 server has more hardware resources than the T330 server. However, in some experiments, the overhead from T340 was higher than the overhead from T330. Consequently, it is important to highlight that not all physical servers have the same suitability for consolidate machines through virtualization.

From the perspective of the server devices, if the physical resources are different, the results for the consolidation overhead should be different as well. However, since real measurements are needed, the proposed method could be applied in the same manner. We selected the workload for the CPU workload to illustrate the usefulness and applicability of the proposed method. However, the proposed method could be applied for any kind of workload.

Finally, in this work, devices such as RAM and disk space were not considered. Nevertheless, the proposed method could be applied in the same manner by selecting a suitable workload and monitoring the mean response time. Although the selection of a different device and workload would generate different consolidation overhead values, the method could still be applied for any configuration.

9 Conclusions and future work

In this work, a method for quantifying server consolidation overhead was proposed. This method can be applied to any consolidation scenario, regardless of the virtualization technology implemented (VMs or containers). We proposed a simple quantification for OV_v and OV_c, obtaining a recursive description. On this basis, a recursive algorithm was implemented to automate the overhead quantification process.

To demonstrate the usability and applicability of the proposed method, a large set of experiments were performed in a real environment. We varied the consolidation degree, the hypervisor (Type I, Type II, and container-based), the combination of VMs and containers, and the workload demand. In all experiments, the executed workload was CPU intensive, and the mean response time was measured and calculated for each case. As an opposing metric, we also calculated the percentage of time that the physical server performed useful work despite the consolidation overhead.

Table 5 Mean response time for each scenario

N	Scenario	R	N	Scenario	R
1	A	27.70	5	J	6.14
2	B	16.77	5	K	9.81
2	C	17.51	5	L	7.20
3	D	14.29	6	M	7.73
3	E	12.56	6	N	7.53
4	F	12.31	6	O	3.59
4	G	8.45	6	P	7.53
4	H	8.59	6	Q	3.90
5	I	9.67	6	R	5.08

As a result, we can conclude that OV_v and OV_c depend mainly on the degree of consolidation and the combination of VMs and containers. Generally, the higher the consolidation degree was, the greater the OV_c value. Additionally, OV_c was higher for Type II and container-based hypervisors due to the number of software layers, whereas lower numbers of containers per VM resulted in lower OV_c values.

Therefore, the research question presented at the beginning of the paper has been answered by proposing a general method for quantifying the consolidation overhead regardless of the consolidation scenario and studying the values of OV_v and OV_c for a broad spectrum of scenarios. This information can help system administrators make more suitable decisions regarding server consolidation to minimize OV_v and OV_c and maximize the percentage of useful work time.

In future work, we will consider implement the proposed method in an environment based on different amounts of memory or I/O workloads. Additionally, it would be interesting to apply the method to other commercial hypervisors and to monitor performance metrics according to the amount of RAM and hard disk space in order to identify which resources are affected by the different hypervisors. Additionally, the application of different cloud workloads such as CloudSuite and Intel Hibench could be considered.

Regarding the heterogeneity of virtual servers, we could consider executing different workloads in each virtual server. Furthermore, it would be interesting to identify a way to compare the proposed method with existing techniques. Moreover, developing an automatic tool to quantify and represent the values of OV_v and OV_c would be very useful for system administrators.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Bachiega NG, Souza PS, Bruschi SM, De Souza, SDR (2018) Container-based performance evaluation: a survey and challenges. In: 2018 IEEE international conference on cloud engineering (IC2E). IEEE, pp 398–403
2. Bermejo B, Juiz C (2021) On the classification and quantification of server consolidation overheads. J Supercomput 77:1
3. Bermejo B, Juiz C, Guerrero C (2019) Virtualization and consolidation: a systematic review of the past 10 years of research on energy and performance. J Supercomput 75(2):808–836
4. Bhardwaj A, Krishna CR (2021) Virtualization in cloud computing: Moving from hypervisor to containerization—a survey. Arab J Sci Eng 58:1–17

5. Casalicchio E (2019) A study on performance measures for auto-scaling cpu-intensive containerized applications. Clust Comput 22(3):995–1006
6. Chae M, Lee H, Lee K (2019) A performance comparison of linux containers and virtual machines using docker and kvm. Clust Comput 22(1):1765–1775
7. Desai PR (2016) A survey of performance comparison between virtual machines and containers. Int J Comput Sci Eng 4(7):55–59
8. Efoui-Hess M (2019) Climate crisis: The unsustainable use of online video. The Shift Project: Paris, France
9. Helali L, Omri MN (2021) A survey of data center consolidation in cloud computing systems. Computer Sci Rev 39:100366
10. Huber N, von Quast M, Brosig F, Hauck M, Kounev S (2011) A method for experimental analysis and modeling of virtualization performance overhead. In: International conference on cloud computing and services science. Springer, pp 353–370
11. Kleinrock L (1967) Time-shared systems: A theoretical treatment. J ACM (JACM) 14(2):242–261
12. Mardan AAA, Kono K (2020) When the virtual machine wins over the container: Dbms performance and isolation in virtualized environments. J Inf Process 28:369–377
13. Martin JP, Kandasamy A, Chandrasekaran K (2018) Exploring the support for high performance applications in the container runtime environment. HCIS 8(1):1–15
14. Molero, X., Juiz, C., Roden˜o, M.: Evaluaci´on y modelado del rendimiento de los sistemas inform´aticos. Pearson Educaci´on London (2004)
15. Xu F, Liu F, Jin H, Vasilakos AV (2013) Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions. Proc IEEE 102(1):11–31