The Journal of Supercomputing
https://doi.org/10.1007/s11227-024-05943-y

On the scalability of the speedup considering the overhead of consolidating virtual machines in servers for data centers

Carlos Juiz[1] · **Belen Bermejo**[1]

Accepted: 27 January 2024

Abstract
Virtualization technologies are extensively utilized in data centers, particularly cloud computing. This facilitates data center management and diminishes the number of physical machines (servers) and, subsequently, their cooling requirements, leading to cost, space, and power consumption reductions. When applications in data centers are executing independent parallel transactions, but with similar performance requirements, the appropriate level of virtual machine consolidation on a server poses a fundamental challenge for capacity planning. This article introduces a method to evaluate the performance speedup achieved through virtualization on any server and the effects of virtualization and consolidation overheads on physical or virtual machine scalability. This research formalizes the speedup and overheads, using classical computer architecture statements. but at the same time proposes a new method to analyze these overhead amounts and types, showing the scalability and efficiency of different consolidations in the same server and its comparison against no consolidation. This work also proposes a new way to determine the optimal number of physical servers and the optimal number of consolidated virtual machines for a given transaction workload. The real experimentation was performed with different workload sizes, types of virtualizations and different servers. The method presented also facilitates the representation of linear scalability against the real degree of parallelism of either physical machines or consolidated virtual machines for a given transaction workload, as well as striking the right balance between speedup and energy in virtual server consolidation.

The authors Carlos Juiz and Belen Bermejo have contributed equally to this work.

✉ Carlos Juiz
cjuiz@uib.es

Belen Bermejo
belen.bermejo@uib.es

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

Published online: 15 February 2024

Keywords Consolidation · Virtualization · Servers · Speedup · Scalability · Performance evaluation

1 Introduction

Cloud computing and data centers have become integral parts of our daily lives, thanks to their scalable services [1]. Data centers offer users the illusion of unlimited resources, making them highly valuable. Virtualization technology is vital for cloud providers, enabling them to utilize resource multiplexing, migration, server consolidation, and virtual machine resizing [2]. These features empower cloud providers to deliver on-demand resources to their users and customers. However, the ever-increasing demand for resources necessitates providers to find ways to reduce data center space, electrical power, cooling, and ultimately, the costs associated with hosting numerous physical machines (PMs) within them. One approach to address this challenge is the server consolidation technique, wherein multiple virtual machines (VMs) are bundled into the fewest possible number of PMs.

Despite the mentioned advantages, virtualization comes with a trade-off as it can potentially degrade server performance due to the additional overhead inherent in virtualization software and when handling multiple consolidated VMs and their workload-sharing, even when the executing tasks have no logical interdependence or communication among them [3]. As a result, virtual server consolidation aims to optimize data center resource utilization by packing several VMs into some PM. However, it is essential to strike a balance by considering the number of VMs per PM to avoid overexploitation of virtualization, known as VM *sprawling*. This phenomenon occurs when VMs proliferate uncontrollably in data centers, leading to increased management and maintenance complexities. VM sprawling is often caused by factors like overprovisioning, lack of proactive VM allocation maintenance, or uncontrolled migration [4].

Virtualization offers enhanced data center management flexibility by introducing a layer of virtual machine management (VMM) or hypervisor. The hypervisor creates and manages virtual machines, that can operate in isolated execution environments. There are two main approaches to implementing virtualization, from a system point of view (directly on top of hardware) or a process point of view (requiring an operating system for deployment). Indeed, the introduction of the hypervisor in virtualization adds overhead to the system. The extent of this overhead varies depending on how virtualization is implemented, and different approaches can result in different magnitudes of overhead [5]. Moreover, when virtual machine consolidation occurs, an additional type of overhead is introduced. As the number of consolidated virtual machines increases, issues similar to those found in parallelism, such as consistency and communication, come into play, even though the workload tasks to be executed in different machines would be independent. The level of consolidation directly impacts the extent of performance degradation due to the overhead, which in turn affects the quality of service experienced by users. Various factors influence the specific value of VM consolidation overhead. These factors include the type of hypervisor technology

utilized, the workload being executed, and other relevant aspects [3]. As the number of virtual machines consolidated within the same physical machine increases, the overhead introduced by the hypervisor also increases. The hypervisor is responsible for managing all the VMs concurrently, which means it needs to allocate and manage resources for each VM, leading to increased resource demands and contention. The overhead arises from the hypervisor's additional tasks, such as scheduling, memory management, and handling requests for multiple VMs. The more VMs there are on a single PM, the more complex and resource intensive these management tasks become, leading to performance degradation.

Thus, in virtualized data centers, consolidation is the process of allocating VMs into the smallest number of PMs to improve resource utilization. Once the VMs are consolidated into fewer PMs, the idle PMs can be shut down or suspended for savings of different indicators, mainly power consumption. Finding the right balance in VM consolidation is crucial to avoid overloading the hypervisor and ensure efficient resource utilization. System managers need to carefully assess the workload requirements and performance characteristics to determine the optimal degree of consolidation that minimizes overhead while maximizing resource utilization and server performance.

1.1 Contribution

The quantification of overhead for various configurations of virtual servers is an essential step in understanding the performance implications of virtualization. Once this magnitude is known, the next critical aspect is to determine the optimal degree of VM consolidation in a PM, which reflects the scalability of consolidation. Our article presents a method to evaluate the speedup scalability (and energy efficiency) achieved through consolidating any server with virtualization and consolidation capabilities. The proposed method not only enables the assessment of scalability and efficiency for different consolidation scenarios on the same server, but also benchmarks different servers.

Additionally, while there are de jure standardizations for physical servers in data centers, such as determining their performance and energy efficiency, there is a lack of standardization regarding this extent to virtualization. For example, the ISO/IEC 30134-4 defines a method to measure the peak capacity (server saturation) and utilization of servers operating in a data center using operator-selected performance benchmarks. However, this standard does not provide a method for comparing individual server energy effectiveness across data centers. To provide a method for comparing individual server energy effectiveness, the ISO/IEC 21836 standard provides a metric to measure and report the energy effectiveness of specific server configurations. However, even though consolidation is a traditional green IT technique, deployed in most data centers to reduce the number of turn-on equipment and to increase the utilization of the remaining servers, performance is degraded due to the added consolidation software and the competition for hardware resources among consolidated virtual machines. Specifically, there is no established method to formalize the speedup scalability of servers for comparatively determining either

the optimal number of physical machines or the appropriate consolidation of virtual machines into a single physical machine for a given workload of parallel independent but similar tasks, and less considering performance and energy together, using speedup and efficiency. This knowledge is crucial for having clear guidelines on virtualization levels for a given server to significantly benefit the overall efficiency and management of data centers. This paper focuses on the system's point of view, with particular attention to system virtual machines and different virtualization techniques: full virtualization, para virtualization, and hardware-assisted virtualization. Additionally, it examines the two types of hypervisors, namely type I and type II, while not upon containers (see Fig. 1). For brevity, most experiments will concentrate on type I hypervisors, although the findings can be generalized to type II hypervisors as we shall illustrate.

Therefore, the scalability is not only assessed when the workload is increasing but also evaluated when the number of virtual machines is consolidated in the same physical machine for a given workload of independent but similar tasks. This data center configuration in time division or space division virtualization is usual in e-commerce applications. For example, in the tourism industry for making reservations, seat room availability or price yielding, independent transactions of similar performance requirements in CPU and memory are divided in parallel virtual machines consolidated in one server [7].

Thus, the research questions in this research are the following:

- RQ1: How do the speedup and the efficiency of nonconsolidated servers (PMs) versus consolidation of virtual machines in one server (VMs) parallel scenarios behave for a given workload due to the VMs overheads?

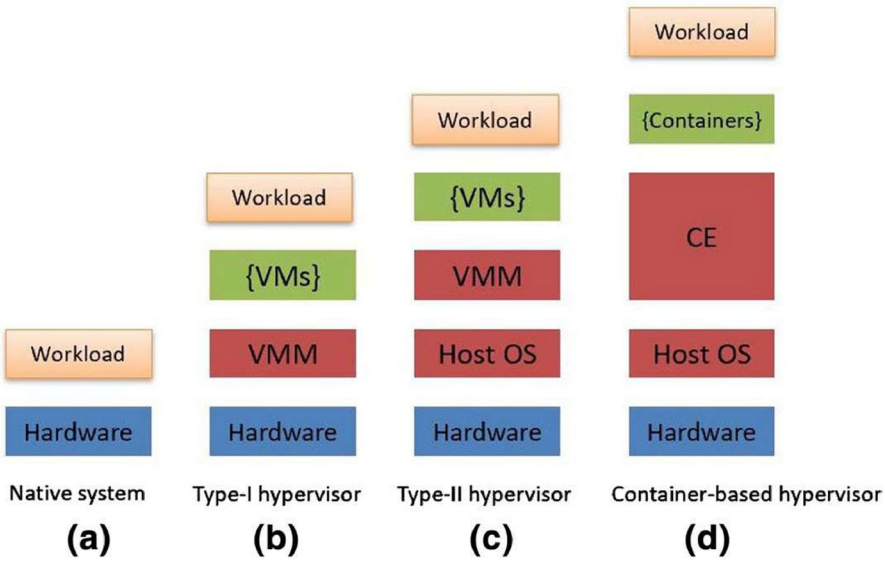

Fig. 1 Types of virtualizations and sources of system overheads [6]

- RQ2: What is the relationship between the measured speedup and the efficiency of the number of machines considered (PMs vs. VMs) with the linear scalability?
- RQ3: Is it possible to estimate the optimal number of VMs in consolidation for any type of server with some consolidation efficiency indicator?
- RQ4: How to depict, in a consistent and coherent representation, the speedup in comparison with linear scalability to easily detect, in a view, the optimal number of parallel PMs and the optimal number of consolidated VMs for a given workload, and even suboptimal scenarios as a result of the VMs overheads?
- RQ5: Is this representation method able to point out different regions for nonconsolidation, consolidation and server sprawling, respectively, in comparison with the linear speedup pattern?
- RQ6: Is this proposed representation applicable to compare these respective scenarios for their performance-energy trade-off, measured with one joint indicator of speedup and energy efficiency?

1.2 Article structure

This article is organized as follows. Section 2 overviews the work related to the research topic and the novelty of this paper. Section 3 is devoted to the formalization of the investigated problem, the proposed theoretical formulation of the applicability of the classic performance concepts in computer architecture, including speedup and efficiency, the less-known isoefficiency, and how to introduce the virtualization and consolidation overheads in this classical formulation (answering to RQ1). Section 4 is the problem statement, providing numerical and graphical examples to overview the speedup scalability state of the problem in determining the optimal number of either physical machines or consolidated virtual machines for a given transaction workload and the problem of the nonlinear shape of speedup scalability when consolidating VMs. Section 5 is devoted to speedup scalability and its variable linearity (answering RQ2). Section 6 proposes some heuristics to overcome the problem of determining the optimal degree of consolidation per server (answering RQ3). In Sect. 7, through elementary trigonometry, the linearity constant is proposed as a reference to determine the nonconsolidation and consolidation intervals and also as a consistent and coherent graphical representation to compare consolidation scenarios based on performance and even jointly with energy. This new graphical reference permits the formal determination of either using the optimal number of physical machines (and suboptimal) or using the optimal number of virtual machines (and suboptimal) for a given transaction workload in a server (answering RQ4). Additionally, the new graphical representation determines three different speedup (and energy) scalability regions for a given server the one where consolidation may not be efficient due to virtualization overheads, the one where consolidation starts to become more efficient because parallelism through virtual machines pays, and the last one in which the physical or virtual machines sprawling discourages to split more the workload into smaller independent tasks for a given server (answering RQ5 and RQ6). Section 8 shows the empirical results from previous theoretical

formalization and their practical applicability. It delves into the applied methodology and explains how the experiments have been carried out with different servers and different workload amounts, type I and type II virtualizations and performance energy trade-off indexing. The article ends with the discussion Sect. 9, including theoretical implications, regarding their relationship with Amdahl, Gustafson and USL laws, and practical implications, including the limitations and future work. Conclusions are in Sect. 10.

2 Related work, motivation and novelty

Various studies have been conducted to explore the impact of consolidation overhead and performance degradation on consolidated systems and the quality of service reviewing extensively the literature [8]. Some recent works such as [9] present a comprehensive analysis of cloud computing virtual machine consolidation, exploring various strategies, benefits, challenges and future trends in this domain. The authors stated that consolidating virtual machines for cloud computing can be difficult, since it is complex to achieve the right balance between energy usage, resource usage, and service quality demands. Also, the challenge arises due to the dynamic nature of cloud workloads and the varying resource demands of different applications. The fundamental issue with VM consolidation solutions is the trade-off between energy efficiency, QoS and optimum SLA Violations. The research work [10] proposes multiple resources aware VM consolidation, a unique approach for the dynamic VM consolidation framework in the cloud data center. The paper [11] presents a taxonomy comprising resource assignment methods, metrics, objective functions, migration methods, algorithmic methods, co-location criteria of VMs, architectures, workload datasets, and evaluation approaches in VM consolidation. The authors also reviewed related work regarding the resources of PMs, algorithms methods, metrics, architectures, and the objectives in static/dynamic VM consolidation. In their research [12], the authors thoroughly investigated influential factors that impact the performance of consolidated servers. They also conducted a comprehensive review of state-of-the-art research on managing virtual machine performance overhead and delved into the underlying causes of such overheads. Building upon the insights gained from [12], the authors of [5] classified the various types of consolidation overhead and introduced a general method to estimate execution times, which includes accounting for these overheads. This classification and estimation method is versatile enough to be applied to systems with different server configurations and workload characteristics, encompassing virtual machines and containers. By adopting a holistic perspective that focuses on the physical server and considering all aspects of virtualization, their work contributes to a more comprehensive understanding of the performance implications of consolidation and provides valuable insights for optimizing data center resource utilization and system management. Subsequently, in [6], the execution time estimation method was further extended to consider nested combinations of successive consolidation levels, particularly containers within virtual machines, which, in turn, are hosted on physical machines. This is just a selection of articles, relevant in impact and citations,

however does not take speedup or scalability into account for the consolidation of virtual machines.

Additionally, to VM consolidation, in our research, we revisit and reinterpret fundamental concepts of parallelism, including speedup, efficiency per machine, the efficiency of execution time, isoefficiency, and introduce new concepts specifically tailored to the consolidation of virtual machines in physical machines. This endeavor is not without its risks, as reinterpreting basic concepts may make it challenging to select relevant related works. The extensive and popular bibliography on the speedup and efficiency subjects adds to the complexity of the related work selection while adding the uncertainty of potential misinterpretations of forgetting some classical references. Due to space constraints, not all pertinent references can be cited in the article. However, the authors suggest that [13–15] might serve as a useful summary of classical definitions of speedup, efficiency, isoefficiency, and the well-known Amdahl's [16] and Gustafson's [17] laws, all brought together in a single interpretation through [18]. The inspiration for this work stems from Gunther's proposal in [19], where he modifies Amdahl's law through his universal scalability law (USL). In addition to this theoretical inspirational motivation, other motivations were originated from practitioners that may want not only to quantify if they comply with the performance SLAs when either adding PMs or consolidating more VMs per PM but also in terms of server scalability and its overhead impact in this dilemma, when they experienced or forecast an incremental workload with independent tasks per machine. A recent survey about consolidating VMs is [8] giving a comprehensive SLR on consolidation solutions, methods and metrics, mainly about performance and energy. However, from our knowledge, no study has focused on the consolidation capability of servers (PMs) in time division virtualization, by formalizing the performance speedup degradation due to overheads, in comparison with the parallelization of PMs (and its consequences on energy); moreover, first using the traditional techniques of speedup representation and compared with the laws mentioned above, and then proposing an alternative way representation mode of speedup against linearity to detect optimal and suboptimal consolidations and nonconsolidations.

Our methodology allows for comparisons among different physical servers and different types of virtualization techniques. Additionally, the method introduces a novel representation of intervals of consolidation feasibility that go beyond considering only speedup but also incorporating performance and energy trade-off indexes. By considering a new format for speedup representation against linearity, this approach offers a comprehensive evaluation of the consolidation process and its implications for system managers but also a new way to represent speedup graphically.

3 Formalization of speedup, overheads and efficiencies

The primary objective of this work is to revisit and reuse fundamental concepts of parallelism and integrate them to determine the optimal degree of VM consolidation in one PM. To define our model, method, examples and experimentation,

we distinguish the following two scenarios. On the one hand let's take N homogeneous physical machines (PMs) that execute in parallel a CPU and memory workload, of total size T_m, but divisible in N independent tasks (each of size T_m/N), for example, transactions to be executed in these N servers; and, on the other hand, let's take one PM, identical to the previous set, that executes the same workload in time division mode, that is divided in the same N independent tasks but in N virtual machines (VMs). Thus, we define (see Fig. 2):

- $T_{\mathrm{PM}}(N)$ as the mean execution time of the tasks of size T_m/N of the workload by the N physical servers (PMs).
- $T_{\mathrm{VM}}(N)$ as the mean execution time of the tasks of the same size T_m/N of the workload by the N virtual servers (VMs).

Intuitively, it is clear that the scenario of having parallel PMs is executing the independent tasks more quickly than the parallel VMs at one PM, i.e., $T_{\mathrm{PM}}(N) < T_{\mathrm{VM}}(N)$, since N parallel PMs are more rapid than one PM even with VM parallelization and also the virtualization is not for free in terms of performance.

The additional workload required to support consolidation exists irrespective of the hypervisor used. Each hypervisor may have different software requirements for implementation, but this article is focused first on describing the essential layers of interest, as we shown in Fig. 1.

3.1 Speedup of PMs over VMs

The speedup factor of the homogeneous PMs over the VMs consolidated in an identical PM can be classically [13, 15] calculated as:

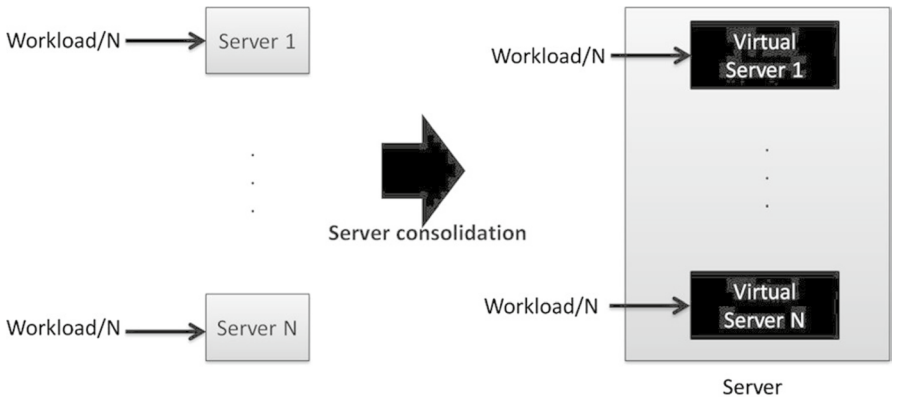

Fig. 2 Independent tasks parallel execution in either homogeneous servers (PMs) or consolidated virtual machines (VMs) [5]

$$S(N) = \frac{T_{\mathrm{VM}}(N)}{T_{\mathrm{PM}}(N)} \tag{1}$$

Parallel homogeneous PMs are faster than a single PM parallelizing by virtualization. The improvement factor will be higher or lower depending mainly on the resources of the servers, relative to the size of the executable tasks of size T_m/N, and the virtualization characteristics of the server. A particular result is the speedup that occurs with a single server due to virtualization software:

$$S(1) = \frac{T_{\mathrm{VM}}(1)}{T_{\mathrm{PM}}(1)} > 1 \tag{2}$$

Thus, we can define the virtualization software overhead as $OV_v(1)$, necessary to be able to have at least one VM in a PM (although for functional purposes it is not very useful). Expressed through mean execution times:

$$S(1) = \frac{T_{\mathrm{VM}}(1)}{T_{\mathrm{PM}}(1)} = \frac{T_{\mathrm{PM}}(1) + OV_v(1)}{T_{\mathrm{PM}}(1)} = 1 + \frac{OV_v(1)}{T_{\mathrm{PM}}(1)} \tag{3}$$

That is, the mean execution time of the workload in the PM is the pattern on which the speedup is calculated, and $OV_v(1)$ is the overhead of deploying virtualization software in one PM. Thus, the overhead of virtualization shared by N virtual machines would be $OV_v(N) = \frac{OV_v(1)}{N}$.

This calculates the overhead of one VM on one PM across the cluster. This simplification is applicable because all PMs are homogeneous and their VMs are identical (for each VM spawned on the respective PM).

Analogously, we can express the speedup of N PMs over N VMs consolidated in one identical PM, expressed in mean execution times:

$$S(N) = \frac{T_{\mathrm{VM}}(N)}{T_{\mathrm{PM}}(N)} = \frac{T_{\mathrm{PM}}(N) + OV_v(N) + OV_c(N)}{T_{\mathrm{PM}}(N)}, \tag{4}$$

where $OV_c(N)$ is the overhead of having more than one VM consolidated in the same PM. This latest overhead is due to the interactions of VMM management of the VMs and any additional time executing the workload, due to the VMs running in parallel on a single PM, even if the application tasks are independent and have no communication among them. However, the consolidation permits some reduction of total workload execution time due to the software parallelism. Figure 3 shows the mean execution time of independent tasks in a virtual machine, i.e., $T_{\mathrm{VM}}(N)$, in the function of the mean execution time of the same tasks in a physical machine, i.e., $T_{\mathrm{PM}}(N)$, plus the overheads described above.

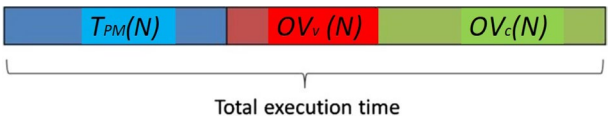

Fig. 3 Decomposition of the time $T_{\mathrm{VM}}(N)$ [5]

Thus, the proposed method in this research defines a decomposed model with three essential mean execution times that apply to $T_{\mathrm{VM}}(N)$:

- Execution time on the equivalent physical machine (PM) or effective workload: This refers to the time it takes for the task to be executed on a physical machine directly, without any virtualization overhead. It represents the baseline performance when there is no virtualization involved.
- Overhead of the virtualization layer: This represents the overhead introduced by the virtualization layer, and its value is distributed among all the virtual machines (VMs) running on the PM. It accounts for the common overhead incurred by any virtualization technology.
- Consolidation machines overhead: This overhead depends on the number of VM consolidated in a PM, meaning multiple virtual servers running in parallel on a single physical server. As the number of VMs increases, this variable overhead could become more significant due to the hypervisor's additional management and resource allocation demands.

The definition of speedup in (4) allows us to express it by separating parts of the speedup:

$$S(N) = 1 + \frac{OV_v(N)}{T_{\mathrm{PM}}(N)} + \frac{OV_c(N)}{T_{\mathrm{PM}}(N)} \tag{5}$$

Expressing the speedup in fractions, we have three parts, the parallel part of useful work (which, as Amdahl law express it improves by 100%, and therefore scales linearly), the fraction corresponding to OV_v, which we will note as γ, and the fraction corresponding to OV_c multiplied by the number of machines, which we will denote as δ:

$$S(N) = \frac{1}{\frac{1}{N} + \gamma + N \cdot \delta} = \frac{N}{1 + N \cdot (\gamma + N \cdot \delta)}, \tag{6}$$

where N are the number of machines in parallel either PMs or VMs.

For the particular case of the speedup of a PM over a VM hosted in an identical PM ($N = 1$), we can solve for γ, since there is only OV_v and there is no OV_c,

$$S(1) = \frac{1}{1 + \gamma} \tag{7}$$

Then,

$$\gamma = \frac{1}{S(1)} - 1 \tag{8}$$

Substituting this value into $S(N)$ at (6), we may calculate δ:

$$\delta = \frac{\frac{\frac{N}{S(N)}-1}{N} - \gamma}{N} = \frac{(\frac{1}{S(N)} - \frac{1}{N}) - \gamma}{N} \tag{9}$$

Expressed with speedups and N machines:

$$\delta = \frac{(\frac{1}{S(N)} - \frac{1}{N}) - (\frac{1}{S(1)} - 1)}{N} \tag{10}$$

The formula of δ vaguely reminds how to determine α in Amdahl's law and even the use of two parameters, γ and δ, reminds α and β in Gunther's USL. However, there is a crucial difference in our formalization. Usually, in Amdahl, Gustafson and USL laws, the speedup or the amount of concurrency (parallel speedup) is diminished by a fraction of contention (α) due to a sequential workload and/or a coherency fraction (β) due to communication among the machines (or processors) considered with the incremental workload. In our model, we are computing the speedup between two scenarios, one of the parallel PMs against one of the parallel VMs consolidated in one identical PM, in a fair comparison by executing the same workload in N independent parallel tasks of the same size T_m/N.

That is, we are comparing the speedup of two systems with the same homogeneous servers, but one scaled by hardware and the other scaled by software with the same amount of workload, but divisible in parallel tasks. The interpretation of the speedup is similar, the overhead diminishes the amount of concurrency of N VMs due to contention (OV_v) of the VMM and the coherency of consolidation (OV_c) in comparison with the amount of concurrency of N PMs.

It is also feasible to express γ and δ with mean execution times. Then, through formulas (3) and (7), we may derive γ in function of mean execution times in one PM and the virtualization overhead:

$$\gamma = \frac{T_{\text{PM}}(1)}{T_{\text{PM}}(1) + OV_v(1)} - 1, \tag{11}$$

whereas δ may also be described with mean execution times and overheads derived from (9) and (11):

$$\delta = \frac{\left(\frac{T_{\text{PM}}(N)}{T_{\text{PM}}(N)+OV_v(N)+OV_c(N)} - \frac{1}{N}\right) - \left(\frac{T_{\text{PM}}(1)}{T_{\text{PM}}(1)+OV_v(1)} - 1\right)}{N} \tag{12}$$

Notice that any software for virtualization will produce that $S(1) > 1$, and then $\gamma < 1$. Let us remember that it is because we established the speedup $S(N)$ as the one produced by the N parallel PMs over the N parallel VMs in a single PM, and the result is $OV_v(1) > 0$. However, $\delta = 0$ for $N = 1$, since for $S(1)$ there is no VM consolidation, that is, $OV_c(1) = 0$. Whereas for $N > 1$ then $\delta \neq 0$, because there is already an overhead of $OV_c(N)$ but at the same time concurrency among virtual machines.

3.2 Efficiency and isoefficiency

Once the speedup $S(N)$ of the parallel PM over the consolidated VMs is defined, the efficiency per PM can be established dividing by the number of machines [13]:

$$E(N) = \frac{S(N)}{N} \tag{13}$$

In the single-machine case, the efficiency of PM over VM is superlinear due to the overhead OV_v:

$$E(1) = S(1) = \frac{1}{1+\gamma} > 1 \tag{14}$$

This is not an anomaly, but a feature when setting the throttling of a PM against that same PM with an overhead (for packing a VM) since it produces inefficiency. In any case, the efficiency for any number N of machines is obtained by dividing their speedup by N:

$$E(N) = \frac{1}{1+N\cdot(\gamma+N\cdot\delta)} \tag{15}$$

Analogously for the mean execution times (see (5) and (6))

$$E(N) = \frac{1}{N} + \frac{OV_v(N)}{N\cdot T_{\mathrm{PM}}(N)} + \frac{OV_c(N)}{N\cdot T_{\mathrm{PM}}(N)} \tag{16}$$

The relationship between the value of the isoefficiency [13] for a number N of VMs consolidated in a PM can be calculated as:

$$C(N) = \frac{\frac{1}{S(N)}}{1-\frac{1}{S(N)}} \tag{17}$$

And therefore, the execution time of the PM can be related to the overheads using,

$$T_{\mathrm{PM}}(N) = C(N)\cdot(OV_v(N) + OV_c(N)) \tag{18}$$

Let's take $T_{\mathrm{PM}}(N)$ as the useful pattern (effective) execution time. OV_v is the useless execution time of overhead, due to adding a software layer that allows VM parallelization. OV_c is the useless execution time attributable to synchronization, communication and other delays to have N parallel VMs in a single PM. We can establish the efficiency of execution tasks of size T_m/N in N machines as the inverse of $S(N)$:

$$E_{\mathrm{VM}}(N) = \frac{T_{\mathrm{PM}}(N)}{T_{\mathrm{VM}}(N)} = \frac{T_{\mathrm{PM}}(N)}{T_{\mathrm{PM}}(N) + OV_v(N) + OV_c(N)} \tag{19}$$

In the preceding formulas, a comprehensive approach was introduced to ascertain and assess the overhead in server consolidations, particularly concerning the consolidation of type I virtual machines. In the case of type II virtualization, the

addition of a software layer further exacerbates the slowdown in mean execution times because CPU and RAM resources are more heavily utilized compared to type I virtualization. The type of virtualization does not hinder the application of the formulation from the earlier sections. The only consideration required is to factor in the new fixed overhead introduced by the operating system in OV_v, which increases the negative value of γ. We shall show this fact in the experimentation.

4 Problem statement of the non-linear speedup in consolidation

From now on, we will illustrate with the proposed formalization from Sect. 3, the problem we are confronted with. In this section, we show the scalability slope with different real examples in one system under test (SUT).

Table 1 shows just an example of the values of the execution times, $T_{\text{PM}}(N)$ and $T_{\text{VM}}(N)$, measured in seconds (for reasons of space, only four significant decimals are shown, but the calculations have been made with eight digits of precision) with workload executed from the Sysbench benchmark and a size of 100K prime numbers ($T_m = 100$ K), in a PM with 16 CPU and 8 GB of RAM.

Once we defined γ representing the fraction of the speedup due to the overhead produced by adding software that allows virtualization, in the example in Table 1, we can see how δ expresses the software parallelization of the VMs consolidated in a PM.

Thus, in addition to the null value when there is only one VM, the δ values are kept low. At the same time, the scalability is slightly superlinear, linear, or slightly sublinear, up to and including $S(5)$, respectively. However, from $S(6)$ it decreases due to the increase in δ, indicating that the VMs parallelism of the tasks in $N > 5$ begins to be interesting, from the point of view of speedup, up to and including $S(10)$. Let's notice that the speedup is around a little more than double, although the independent tasks become proportionally smaller with increasing N. In the end, there is a spike in speedup at $N = 11$, which we will discuss in Sect. 9.

Table 1 Execution times, overheads and fractions of overhead

N	$T_{\text{PM}}(N)$	$T_{\text{VM}}(N)$	$OV_v(N)$	$OV_c(N)$	$S(N)$	γ	δ
1	24.2532	25.4260	1.1728	0.0000	1.0483	−0.0461	0.0000
2	9.2707	19.2885	0.5864	9.4314	2.0805	”	0.0133
3	5.2972	15.8380	0.3909	10.149	2.9898	”	0.0157
4	3.5626	13.9777	0.2932	10.121	3.9234	”	0.0127
5	2.6253	12.5415	0.2345	9.6816	4.7771	”	0.0110
6	2.0386	5.1560	0.1954	2.9219	2.5291	”	0.0458
7	1.5668	3.7863	0.1675	2.0519	2.4165	”	0.0452
8	1.3093	2.8964	0.1466	1.4405	2.2121	”	0.0466
9	1.1296	2.2928	0.1303	1.0328	2.0297	”	0.0475
10	1.0060	2.1759	0.1172	1.0526	2.1629	”	0.0408
11	0.8890	3.1450	0.1066	2.1494	3.5377	”	0.0216

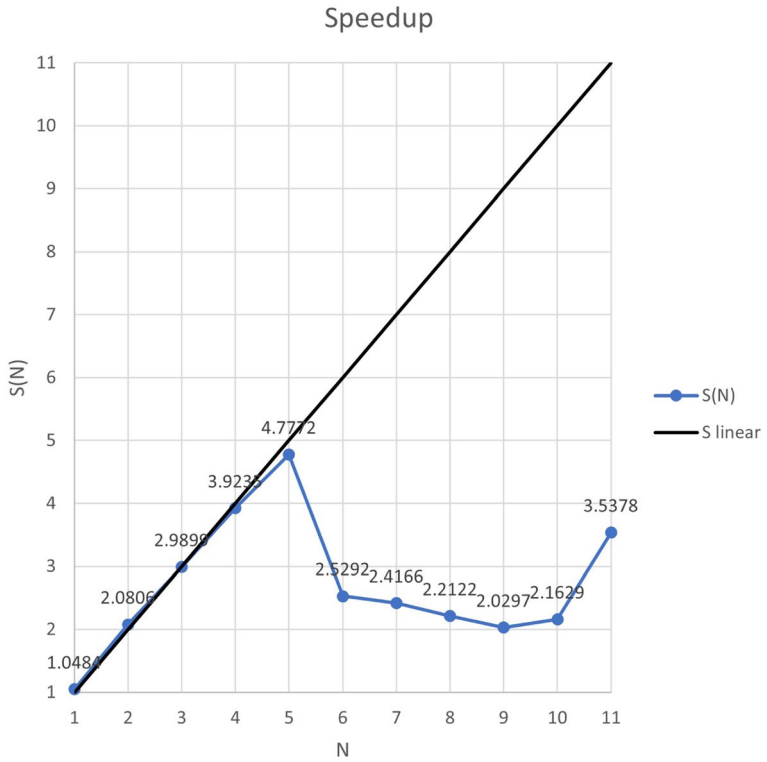

Fig. 4 Speedup of N parallel PMs over N VMs in comparison with linear scalability

These speedup measurements of Table 1 are shown in Fig. 4. It can be seen that initially, the parallel PMs accelerate superlinearly, at first, and almost linearly until $S(5) = 4.7771$, then falling to a valley with speedups between 2.5 and 2.0, where the consolidated VMs become somewhat more efficient, so that the PMs go up again later in $S(11) = 3.5377$, when the workload has been divided into much smaller tasks, in parallel, that is, they are so small that the OV_c already weighs heavily on the $T_{\text{VM}}(N)$. It is very interesting to observe, that the speedup drops compared to N PMs against N VMs, as when dividing the workload of $T_m = 100$ K by $N = 6$, the tasks scale much better than in previous consolidations.

This example illustrates the problem of the nonlinear speedup behavior that we are going develop in subsequent sections to determine when to consolidate or not, from the point of view of that comparative speedup.

To deepen the mentioned problem statement, in Fig. 5, we show different sizes (T_m) of the workload and its incidence in the speedups $S(N)$ for the first nine machines of the example in Table 1. It is evident that the greater the T_m, the greater the linearity of the speedups $S(N)$. Colloquially, we could say that the speedups of the PMs over the consolidated VMs are "stretching" until they

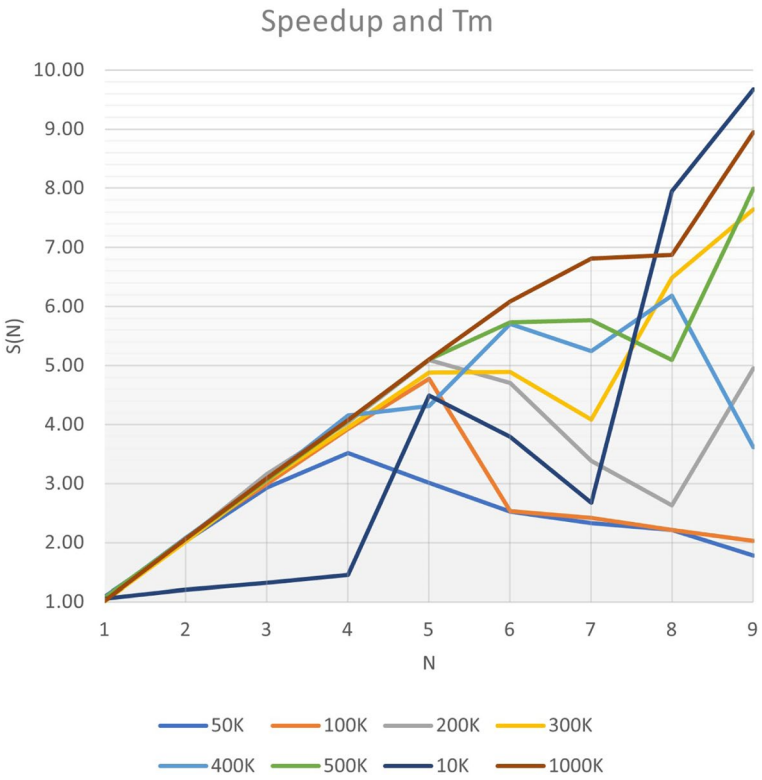

Fig. 5 Speedups of N PMs parallel on consolidation of N VMs varying the size of the problem T_m

Table 2 Speedup, efficiency, isoefficiency and efficiency of the tasks

N	$S(N)$	$E(N)$	$C(N)$	$E_{VM}(N)$	T_{PM}(comp.)
1	1.0483	1.0483	20.6797	0.9538	24.2532
2	2.0805	1.0402	0.9254	0.4806	9.2707
3	2.9898	0.9966	0.5025	0.3344	5.2972
4	3.9234	0.9808	0.3420	0.2548	3.5626
5	4.7771	0.9554	0.2647	0.2093	2.6253
6	2.5291	0.4215	0.6539	0.3953	2.0386
7	2.4165	0.3452	0.7059	0.4138	1.5668
8	2.2121	0.2765	0.8249	0.4520	1.3093
9	2.0297	0.2255	0.9711	0.4926	1.1296
10	2.1629	0.2162	0.8599	0.4623	1.0060
11	3.5377	0.3216	0.3940	0.2826	0.8890

reach linearity with larger task sizes, then "N PMs are N times quicker than one PM". However, this is not the case for all workloads when we consider VM consolidation.

The formalization shown in Sect. 3 permits even to focus on the problem of nonlinearity. For example, Table 2 shows the speedup, efficiency per PM, isoefficiency, and efficiency of the execution tasks for the same example of Table 1. The T_{PM} is the computed execution time, based on the overheads modeled, as a demonstration of the correct calculations done with the isoefficiency formula (18).

In the context of virtualization, consolidation can be considered equally scalable for a given number of machines N if the efficiency $E(N)$ remains constant. However, as shown in Table 2, the efficiency (and consequently $C(N)$) is not constant, indicating that the consolidation may not be equally scalable under certain conditions.

The most remarkable aspect of consolidation is how the speedup behaves. Speedup, and consequently the efficiency per PM and the efficiency of executing tasks within VMs, can vary significantly based on the number of VMs consolidated and the workload characteristics. As the number of VMs increases, the overhead from managing multiple VMs concurrently can impact the speedup and overall efficiency, potentially leading to nonlinear or diminishing returns in performance improvement. The "sawtooth" graph pattern shown in Fig. 4, depicting the speedup of parallel PM with VMs consolidated in an identical PM, is a common behavior in such scenarios. As the number of N increases, the speedup experiences a series of local maxima and minima, influenced by the size of the workload divided into N parallel tasks, i.e., T_m/N.

Therefore, even the formalization, in Sect. 3, of the speedup and efficiencies are the primary results of this paper, showing how the speedup and the efficiency of nonconsolidated servers (PMs) versus consolidation of virtual machines in one server (VMs) parallel scenarios behave for a given workload due to the VMs overheads, there are several gaps to solve in next sections. Most of them are related to determining the optimal number of either PMs or VMs and even the sprawling region. Observing the examples, we provided in Sect. 4, the main questions that will be answered in the next sections will be:

- What is the relationship between the measured speedup and the number of machines considered with the linear scalability?
- Is it possible to estimate the optimal number of VMs in consolidation for any server?
- Is there a way to represent for easily detecting the optimal number of parallel PMs and the optimal number of consolidated VMs for a given workload, and even suboptimal scenarios?

5 PMs and VMs scalability behavior

Not remarkably, in linear scalability, the cartesian values of speedup $S(N)$, N, and the origin, form a right triangle, where the arctangent of the acute angle (in radians) is equivalent to the efficiency $E(N)$. In other words, the angle formed by the speedup and N can be interpreted as the efficiency of the consolidation for a specific number of VMs. This geometric representation helps to visualize how the efficiency changes

with the number of VMs and the speedup. As the angle varies, it indicates the varying efficiency at different levels of VM consolidation.

$$\Theta = \arctan \frac{S(N)}{N} = \arctan(E(N)) \tag{20}$$

Transforming the angle Θ of the arctangent of the efficiency from radians to degrees, it becomes visible how the linearity of the speedup $S(N)$ changes with different consolidations of N VMs.

$$\Theta^o = \arctan(E(N)) \cdot \frac{180}{\pi} \tag{21}$$

This angle $\Theta°$ determines the linear quality of the consolidation due to the value of the efficiency of the PM. In this way, we can classify consolidations into two main regions:

- If $\Theta° \geq 45°$, the parallel N PMs are more efficient than the consolidation of N VMs, since $E(N) \geq 1$. So, the N PMs accelerate linearly, and even superlinearly over the consolidated VMs. Compared to N PMs, the T_m/N problem size of independent tasks is still too large to run in parallel with good efficiency on N VMs.
- If $\Theta° < 45°$, the parallel PMs are efficient but less than the previous case when increasing N, since their speedup is sublinear, i.e., $E(N) < 1$. Consolidate N VMs can be an alternative solution to parallelize PMs, that is, to parallelize independent tasks of size T_m/N, in terms of $S(N)$ and $E(N)$.

6 Optimal consolidation through heuristic efficiency functions

Observing these two example cases from the problem statement, setting the optimal number of PMs in parallel and the optimal number of VMs consolidated at one PM involves finding a delicate balance, considering the following two key considerations:

- Balancing resource utilization and overhead: the decision on how many PMs to run in parallel depends on the workload size T_m and the available resources. Running more PMs in parallel may lead to potentially higher performance. However, it also brings nonfunctional disadvantages, such as increased costs, space, and power consumption. On the contrary, as the number of VMs increases, so does the consolidation overhead, which may result in longer execution times and higher energy consumption [20].
- Avoiding sprawling and management complexity: using more consolidated VMs than necessary can lead to sprawling in data centers, where the uncontrolled proliferation of VMs results in additional management and maintenance challenges. This can lead to reduced efficiency and increased complexities in managing the virtualized environment.

Therefore, fine-tuning the consolidation strategy based on these analyses will help in achieving the most efficient and effective virtualization deployment.

We define N^* as the minimum number of PMs that produce the maximum speedup, $S(N^*)$ (local maximum), over their corresponding consolidated virtual VMs. Therefore, N^* is the extent to which it could be more interesting to deploy parallel PMs and thus divide the workload into tasks, given a problem of size T_m in N^* PMs. Thus, N^* corresponds to the parallelism limit of the PMs to reduce the mean execution time per machine, by dividing T_m by N^*, with optimal performance, space and power to consume at data centers. There can be higher local maxima $S(N)$ but at the cost of lower efficiency.

The process to determine N^* would be to find the local maximum that still accelerates almost linearly, or what is the same, its efficiency per PM is $E(N) \approx 1$. Let's note that with very large T_m this can lead us to select a high N^* (large number of parallel PMs). Consequently, to find out how far to parallelize with PMs versus VMs for a given server, it is not enough just to observe the speedup, but to take the local maximum, with the minimum number of machines, for which $E(N)$ is useful.

Conversely, to select the optimal VM consolidation, one could choose the minimum speedup, since it is when parallel PMs are less efficient than parallel VMs. However, enough machines should be selected for optimal efficiency, but not too many VMs so as not to cause sprawling.

For this reason, we have defined the heuristic indicator that we have called *eficonsolidation* $EC(N)$:

$$EC(N) = \frac{E(1) - E(N)}{N} = \frac{N \cdot S(1) - S(N)}{N^2} \tag{22}$$

Thus, we define N^+ as the optimal number of VMs to consolidate since $EC(N^+)$ maximizes the difference between speedups with overhead and minimizes the number of consolidated machines.

$$EC(N^+) = \max \frac{E(1) - E(N)}{N} \tag{23}$$

7 Redefining linear scalability views

The values of N^* and N^+, which we have defined in the previous section, estimate how far to parallelize N^* PMs or optimally consolidate N^+ VMs, respectively. However, both heuristics allow system managers to select several machine values for both situations.

In any case, the following questions remain:

- If N^* is the number of PMs where the speedup is maximum for an $E(N) \approx 1$, how one does determine how close should we get to an efficiency of 1?
- If N^+ is the optimal number of consolidated VMs, considering the eficonsolidation indicator $EC(N^+)$, which maximizes the difference between speedups with

fixed (γ) and variable (δ) overhead factors, and minimizes the number of consolidated machines, is there another better indicator than the eficonsolidation value?

Our goal in this section is to develop a coherent and consistent criterion for determining both N^* and N^+ , while considering the optimal values $E(N)$ and $EC(N)$. Additionally, we aim to graphically represent the quality of consolidation options simply and intuitively, exploring an alternative representation apart from the cartesian axes $S(N)$ and N in comparison with the linear scalability. This new representation should enable us to simultaneously ascertain N^* and N^+ , while assessing the quality of other consolidation options through consolidation intervals.

To achieve this, we propose the utilization of a novel graphical representation. This representation will aid in identifying the ranges within which N^* and N^+ fall, enabling us to comprehend the trade-offs between performance and resource utilization for various N values. Through this plot method, we will gain valuable insights into determining N^* and N^+ simultaneously, without altering the existing formulation. Furthermore, this graphical approach will also provide a simple and clear depiction of the linear quality of different consolidation options, allowing us to make well-informed decisions based on performance intervals. It enables us to establish a coherent and consistent criterion for determining N^* and N^+, while visually assessing the quality of consolidation options with a higher level of clarity and understanding.

7.1 Constant of linearity

We recall that $\Theta°$ represents the angle formed by the line segment connecting the origin to the point ($S(N)$, N) for a given number of N of PM machines, in comparison to N VMs consolidated in an identical PM. If we consider linearity as a reference, where $S(N)$ equals N, then $\Theta°$ will be 45° degrees. As a result, for any number of machines N, there exists a right triangle with equal legs of natural length n (i.e., $S(n) = n$) and a hypotenuse $M(n)$, as shown in Fig. 6.

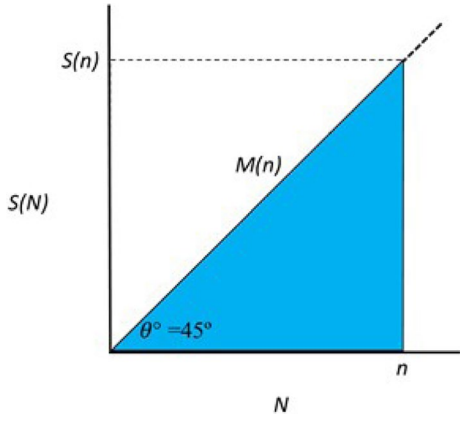

Fig. 6 Right triangle with equal legs of lengths $n = S(n)$ and hypotenuse $M(n)$

Consequently, the well-known relationship in Euclidean geometry (Pythagorean theorem) between the three sides of a right triangle, that is, between the hypotenuse and the equal legs is:

$$M(N) = \sqrt{(S(N))^2 + N^2} = \sqrt{N^2 + N^2} = \sqrt{2} \cdot N \tag{24}$$

Known by many mathematicians as the Pythagorean constant [21], it results from dividing the hypotenuse by the number of machines we consider:

$$\frac{M(N)}{N} = \sqrt{2} \tag{25}$$

In this research, we designate this well-known irrational number as the linearity constant, since it emerges in parallelism when the speedup is linear ($S(N) = N$). As a consequence, this constant serves as a pattern of linear speedup, allowing us to establish a relationship between the ratio $M(N)/N$, representing different speedups of servers, and the ideal right triangle formed by the origin, N, and ($S(N)$, N).

By analyzing the ratios $M(N)/N$, we can establish intervals that determine whether it is more advantageous to consolidate N machines or not. This comparison is made concerning the linearity constant, where N physical machines would precisely achieve N times faster performance than N consolidated virtual machines ($\Theta° = 45°$). By leveraging the linearity constant as a guiding principle, we can effectively assess the effectiveness of consolidation options and align our decisions with our performance objectives.

We may continue with the example we have shown in Fig. 4. We represent the different $M(N)/N$ ratio values are represented compared with the linearity constant dotted red line. As can be seen, in Fig. 7, there are several clearly defined areas:

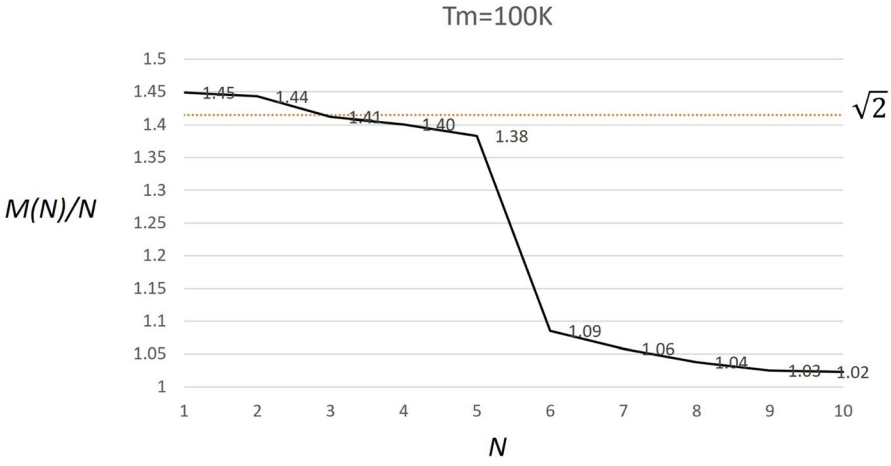

Fig. 7 Linearity ratio ($M(N)/N$) of speedup of N PMs over N VMs in comparison with the constant of linearity for $T_m = 100$ K

- $N \leq 2$, the speedup of N PMs over N VMs is superlinear, due to the overheads of virtualization and consolidation.
- $N = 3$, the speedup is practically linear, and the overheads are compensated with the size of the tasks to be executed by each machine in parallel, which is three times smaller.
- $N > 4$, the speedup starts to be sublinear.
- $N \geq 6$, the speedup of the N PMs has decreased notably, being able to be candidates for N^+ in any of the configurations.

Therefore, two intervals can be indicated in Fig. 7, one optimal for N PM ($N \leq 5$) and another optimal for N VM (from $N \geq 6$).

7.2 Consistent selection of N^* and N^+

The hypotenuses $M(n)$ shown in Fig. 8 reveal another trigonometric characteristic that can aid in determining N^* and N^+, utilizing the same graphical representation. The length of the hypotenuses, $M(n)$, when n is a natural number, follows a sequence of $n \cdot \sqrt{2}$. Consequently, by subtracting the lengths of consecutive hypotenuses, the difference always results in the Pythagorean constant. The sequence of factor $\sqrt{2}$ generates a series of subtractions of consecutive terms:

$$M(n) - M(n-1) = n \cdot \sqrt{2} - (n-1) \cdot \sqrt{2} = \sqrt{2} \tag{26}$$

This consistent difference of $\sqrt{2}$ between consecutive hypotenuses provides us with a valuable pattern that can be leveraged to identify optimal values for N^* and N^+.

If the difference between consecutive terms in a series involving the hypotenuses $M(n)$ and $M(n-1)$ is greater than $\sqrt{2}$, it indicates that the terms are increasing at a faster rate. This difference between $S(N)$ and $S(N-1)$, which affects the calculation of the hypotenuses, signifies a faster speedup of the N PMs over the consolidated N VM. Conversely, if the difference is less than $\sqrt{2}$, it suggests that the terms increase at a slower rate.

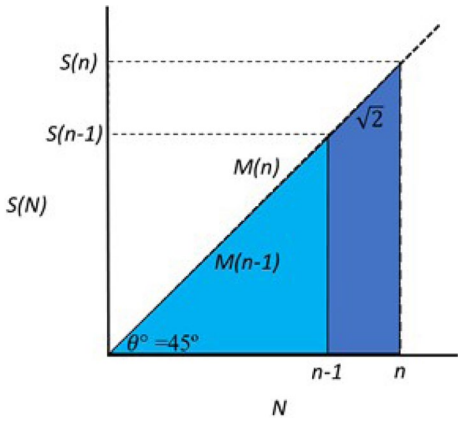

Fig. 8 Right triangles with equal and consecutive legs of natural length

Again, linearity constant $\sqrt{2}$ serves as a crucial basis for comparison between different consolidations. Utilizing this constant not only helps determine the consolidation intervals but also enables the measurement of the relative magnitude of consolidation differences. It provides insights into how the terms of each series differ in their growth or progression, offering a deeper understanding of the consolidation process.

Furthermore, the linearity constant, $\sqrt{2}$, can assist in locating N^* and N^+ as defined previously in the sample, complementing the heuristics of different indicators such as efficiency and eficonsolidation ($E(N)$ and $EC(N)$, respectively). This approach allows for the detection of N^* and N^+ concurrently, coherently and consistently, utilizing the same graphical representation.

Again, the example from the case of Fig. 4 serves as an illustration of this subsection. In Fig. 9 we have kept shown both the linearity constant and the $M(N)/N$ curve of Fig. 8, to now also include the differences values between consecutive hypotenuses. It can be observed that effectively the period where the N PM accelerate linearly or above it, ends in $N = 5$, but now with the hypotenuses differences we show that effectively $N^* = 5$, since it is the value with the maximum speedup and maximum N for the N PMs. We also show that $N^+ = 6$, since the maximum negative difference of the hypotenuses is up to -0.4. From $N \geqslant 7$ the consolidated N VMs are still a good option since the N PMs do not accelerate linearly, although their differences progress little by little (see the differences from 0.89 to 1.01, which are secondary consolidation options for the systems manager).

Let us note that what is shown in the example of Fig. 9 concurses with what was determined with the heuristics explained before, for N^* and N^+. However, this representation is more coherent and consistent. Since it allows using the same pattern, not only to determine the intervals where the N PMs either accelerate linearly or even superlinearly compared to the N VMs or sublinearly but also the magnitude of either the nonconsolidation or the consolidation by determining N^* and N^+, respectively, with the same reference pattern.

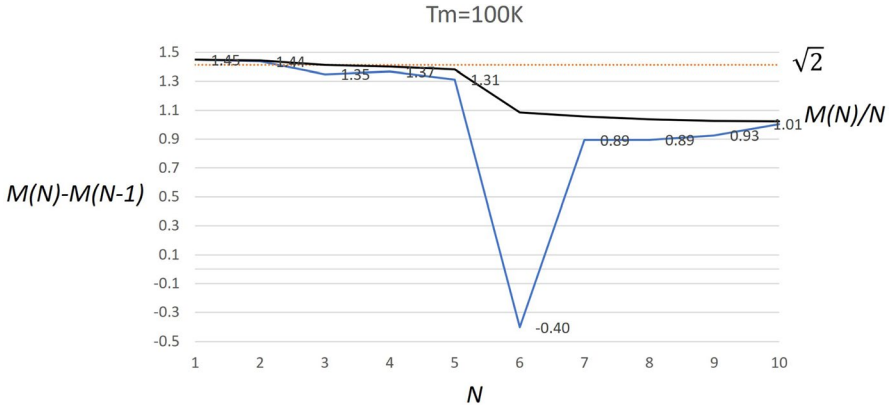

Fig. 9 Linearity ratios (black) and neighbor hypotenuses differences (blue) in comparison with the linearity constant for $T_m = 100$ K

This representation method allows for an immediate visual understanding not only of the consolidation and nonconsolidation intervals but also of the optimal number of machines for both scenarios. By utilizing the linearity constant $\sqrt{2}$ as a reference, we gain valuable insights into the risks associated with certain consolidations. On one hand, it allows us to identify potential issues with sprawling, where consolidation might lead to diminishing returns and decreased performance. On the other hand, it also enables us to recognize the risks of increasing the number of physical servers, with associated problems like increased power consumption and resource waste. In the next lines, we provide a simple algorithm to follow both curves (see Fig. 10).

7.3 Virtual machine consolidation and energy efficiency

One significant advantage of virtualizing and consolidating servers in data centers is the potential for cost savings in hardware, space, cooling, and electrical power consumption. However, it is essential to recognize that as the utilization of PMs increases and CPU saturation is nearly reached, especially under CPU-intensive loads, the benefits of consolidation may diminish. As VMs compete for resources on a highly utilized PM, virtualization overheads and contention can lead to delays in workload execution. These delays may result in increased energy consumption and reduced overall performance, offsetting some of the initial cost savings.

In [20], the CiS^2 metric (*Consolidation Index for CPU-Server Saturation*) was introduced, which takes into account both the performance and energy aspects of N VMs on a single PM in comparison to N parallel PMs executing the same CPU-intensive workload under saturation. The CiS^2 metric can be expressed as the quotient of two values of EDP (Energy-Delay Product) [22]. The EDP metric focuses on the overall energy consumption and performance of a system rather than low-level metrics such as resource utilization or CPU frequency. EDP provides a high-level

```
Algorithm. Selection of N* and N+.
Nmax=set max consolidation number of VMs
Tm=set workload size
Th=set linearity threshold
N*_detected=false
M(0)=0
Max_dif(M(1)-M(0))=false

For N=1 step 1 until Nmax do begin
        Run_benchmark (input:task_size:Tm/N;output:M(N))
        If M(N)/N * Th < √2 and N*_detected= false then
        begin
                N*:=N;
                N*_detected:=true;
                If N*_detected and Max_dif(M(N)-M(N-1)) then
                        N+=N;
        end
end
```

Fig. 10 Selection of N^* and N^+

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. Moreover, EDP is applicable across diverse architectures and technologies, because it allows for comparative analysis across different systems without being biased towards specific low-level server characteristics. This is the reason why, the authors created the CiS^2 metric in [23] combining speedup and energy ratios:

$$CiS^2(N) = \frac{EDP_{\text{VM}}(N)}{EDP_{\text{PM}}(N)} = \frac{E_{\text{VM}}(N) \cdot T_{\text{VM}}(N)}{E_{\text{PM}}(N) \cdot T_{\text{PM}}(N)} = S_e(N) \cdot S(N) \tag{27}$$

Therefore, CiS^2 is the product of the increased ratio in the energy of the consolidated N VMs over the N PM multiplied by their corresponding speedup.

A higher CiS^2 value indicates a less favorable consolidation scenario, where N VMs on a single PM are less energy-efficient and have worse performance and energy consumption, compared to running N parallel PMs with the same workload. On the other hand, a lower CiS^2 value suggests that running N parallel VMs is becoming efficient in terms of energy and performance. Calculating CiS^2 for different values of N determines

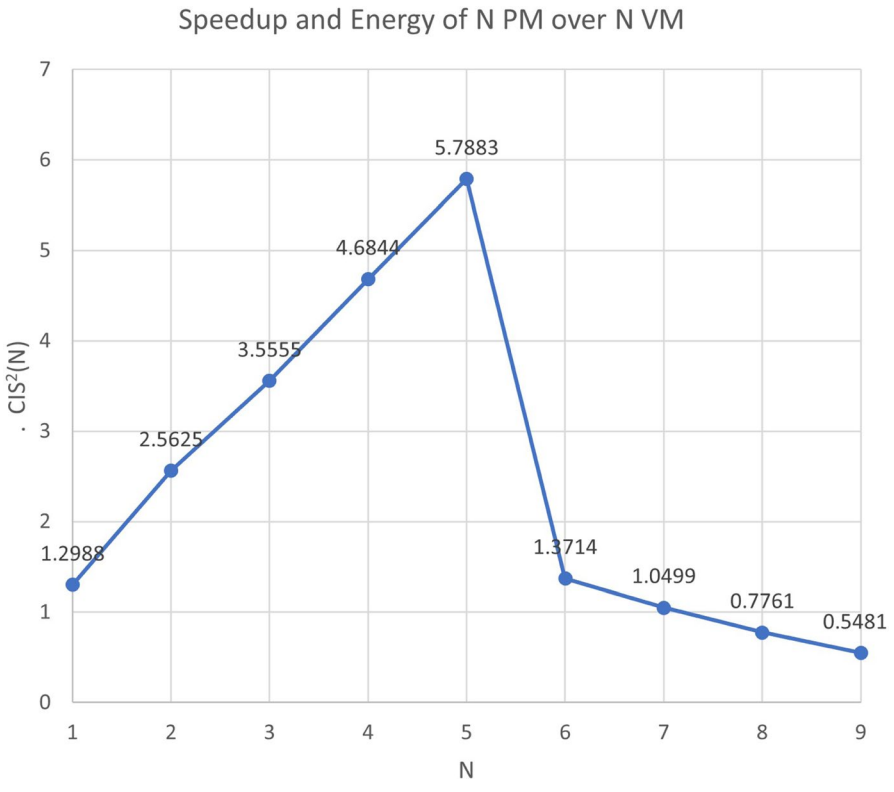

Fig. 11 Performance and energy trade-off $CiS^2(N)$ of N parallel PM over consolidation of N VM

the optimal point where consolidation offers the most significant benefits in terms of energy savings and performance improvement. Thus, the metric permits the evaluation of the efficiency gains achieved through virtualization and consolidation under CPU-server saturation conditions. In Fig. 11, we can see the CiS^2 corresponding (see the similar shape with the example of Fig. 4).

Since energy is the product of power consumption during runtime:

$$CiS^2(N) = S_e(N) \cdot S(N) = \frac{W_{\mathrm{VM}}(N)}{W_{\mathrm{PM}}(N)} \cdot S(N)^2 \tag{28}$$

If we take linear speedup as a reference, that is, $S(N) = N$, in CPU saturation (with CPU workload), the power consumed by N PMs will be N times the power of an identical PM, running the same workload in CPU-saturation. Therefore, substituting $S(N)$ for N:

$$CiS^2(N) = S_e(N) \cdot S(N) = \frac{1}{N} \cdot S(N)^2 = \frac{1}{N} \cdot N^2 \tag{29}$$

Therefore, it can be applied again that for any number of machines n, there exists a right triangle with equal legs of natural length $n = CiS^2(n)$ and hypotenuse $M'(n)$ (see Fig. 11, analogous to Fig. 12). Consequently, applying again the Pythagorean theorem:

$$M'(N) = \sqrt{((CiS^2(N))^2 + N^2)} = \sqrt{(N^2 + N^2)} = \sqrt{2} \cdot N \tag{30}$$

Therefore, when the speedup is linear, with intensive CPU workloads (in CPU saturation):

$$\frac{M'(N)}{N} = \frac{M'(N)}{S(N)} = \frac{M'(N)}{CiS^2(N)} = \sqrt{2} \tag{31}$$

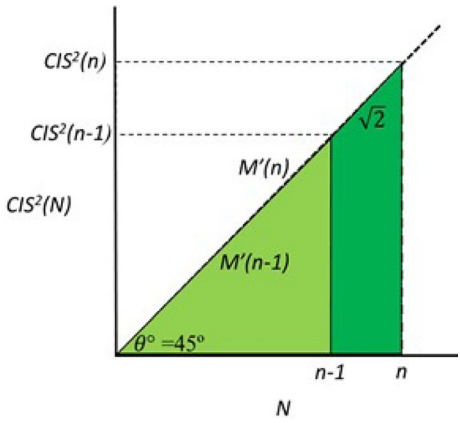

Fig. 12 Two right triangles with equal legs, of lengths $n = CiS^2(n)$ and hypotenuse $M'(n)$, and $n - 1 = CiS^2(n - 1)$, and hypotenuse $M'(n - 1)$, respectively

8 Empirical results

In previous sections, we proposed how to determine overhead, speedup, and efficiencies for server consolidations, specifically, in consolidating virtual machines. Before continuing to explain the rest of the contributions we depict the experimentation setup we performed.

Each system under test (SUT) executes the workload of the benchmark and its behavior is monitored (software and hardware monitors). All interval times are measured in seconds and the electrical power consumed is in watts. We compare the workload execution between the physical machine PM and the consolidated server. We vary the number of VMs that are hosted on the PM, and all the workload is distributed evenly across the set of servers (a time division of workload into equal tasks). Therefore, the overhead in virtualization is calculated by comparing the execution of the workload that is balanced between N physical servers (PMs), with the execution of an identical workload balanced and between N virtual servers (VMs), hosted in the same physical server (see Fig. 2).

To demonstrate the theoretical content of speedup, efficiency, isoefficiency, etc. and the concepts we created, we use an extensive set of values obtained from experiments. The experimental configuration consists offive types of physical servers that use the majority (but not the only) Intel Xeon CPU family: a Dell PowerEdge T430, with 16 physical CPUs, 8 GB of RAM and Ubuntu Server as the operating system, a Dell PowerEdge T330, with 8 physical CPUs, 16 GB RAM and Ubuntu Server 16.04 as the OS, the same as for a 20 CPU Lenovo ST550 with 128GB RAM and a 48 CPU Fujitsu RX5000 with 1024 GB RAM. It has also been experimented with AMD processors, specifically a Ryzen 7. Repeatedly, we have taken the Dell PowerEdge T430 server as an example to illustrate not only the problem to solve, but also to illustrate the concepts and formulation described here for consistency in comparisons between different sizes of the problem and types of virtualizations.

For virtualization, we implement the of use kernel-based virtual machines (KVM) as type I hypervisor and virtual box as type II. All virtual machines are allocated with the same amount of CPU as the physical server, 1 GB of virtual RAM, and the Ubuntu Server as the guest operating system. The workloads executed are from the Sysbench benchmark, which is CPU intensive (calculation of prime numbers) [24]. It is important to note that in this benchmark the executed workload requests 100% utilization of the physical CPU, which represents CPU saturation. All the experiments have been carried out with time division and not spatial distribution of resources [25]. Keep in mind that we are quantifying the overheads of the system as a whole (all its components) under a CPU-intensive execution workload [24]. The execution times of the PMs and VMs have been obtained by performing multiple executions, for all the configurations, resulting in mean execution times with a standard deviation of less than 5%. That is, the experiments were carried out a specific number of times which ensures statistical significance. Since all the experimentation relies on real measurements, some values may vary among experiments even if they have been repeated hundreds of

times to arrive at this standard deviation [26]. Regarding performance and energy measurements, approved power and energy measurement monitors have been used, both by the SPEC consortium [27] and by the ISO standards organization [10], for the measurement of servers. In particular, we measure the power consumption of the SUTs with the Chroma 66200 device.

From Sect. 4, we have used an example to illustrate the entire formulation so that the reader could glimpse the potential of the research carried out and the novelties provided. In this section, we have reserved more experimentation than the examples shown to empirically demonstrate our findings.

8.1 Machines scalability and heuristics of optimal consolidation

Table 3 shows the different values of $\Theta°$ for the consolidations in Fig. 4, as well as the values of $S(N)$, $E(N)$ and δ. We may see the incidence of N in the angle as we move away from the origin of the coordinates.

The angle $\Theta°$ determines the quality of the consolidation due to the value of the efficiency of the PM. In this way we can classify consolidations into two main regions: if $\Theta° \geq 45°$, then parallel N PMs are more efficient than the consolidation of N VMs, since $E(N) \geq 1$; else then $\Theta° < 45°$, the parallel PMs are less efficient than the previous region with increasing N, since their speedup is sublinear, i.e., $E(N) < 1$. Consolidate N VMs can be an alternative solution to parallelize PMs, that is, to parallelize tasks of size T_m/N, in terms of $S(N)$ and $E_{\text{VM}}(N)$.

Table 3 also shows different $EC(N)$ values. Thus, we define N^+ as the optimal number of VMs to consolidate since $EC(N^+)$ maximizes the difference between speedups with overhead and minimizes the number of consolidated machines.

Continuing with Table 3 values, N^+ corresponds to 6 VMs with a speedup of only 2.5292 of 6 parallel PMs over the six consolidated VMs, an angle $\Theta°$ of 22.85°, corresponding to an efficiency of 0.42 and the value of the highest eficonsolidation $EC(N^+)$ of the sample. It is also one of the highest δ values, although not the maximum. There are even higher δ values, and lower speedups, as we have pointed out,

Table 3 Speedup, efficiency per PM, $\Theta°$, δ and eficonsolidation

N	$S(N)$	$E(N)$	$\Theta°$	δ	$EC(N)$
1	1.0483	1.0483	46.3523	0.0000	0.0000
2	2.0805	1.0402	46.1313	0.0133	0.0040
3	2.9898	0.9966	44.9032	0.0157	0.0172
4	3.9234	0.9808	44.4466	0.0127	0.0168
5	4.7771	0.9554	43.6943	0.0110	0.0185
6	2.5291	0.4215	22.8569	0.0458	0.1044
7	2.4165	0.3452	19.0460	0.0452	0.1004
8	2.2121	0.2765	15.4572	0.0466	0.0964
9	2.0297	0.2255	12.7091	0.0475	0.0914
10	2.1629	0.2162	12.2046	0.0408	0.0832
11	3.5377	0.3216	17.8286	0.0216	0.0660

for example, $S(9) = 2.0297$, but at the cost of consolidating 3 additional VMs. However, values close to $EC(N^+)$ are secondary options for a data center manager to consider.

8.2 Varying the PMs and their resources (servers)

When different PMs have distinct resources, such as CPU and memory, their mean execution times tend to differ, although when consolidating in the same manner, utilizing the same benchmark and VMM. However, similarities may arise in the context of N^* and/or N^+ values, leading to a phenomenon we call *isoconsolidation*. Table 4 presents a comparison of PM values with different CPU and RAM resources. We can identify several equal bindings by examining $S(N^*)$, N^*, and N^+.

In Table 4 we show very different servers with different resources in several CPUs and RAM, but most of their values are similar. For instance, consolidating around $N \approx 5$ VMs proves optimal for nearly all PMs except one, given the selected problem size. This observation supports the notion that the benchmark exhibits a greater demand for CPU resources compared to RAM, as evidenced by the nearly identical Dell T430 and T330 servers, except for their inverted resources. Furthermore, the T330 may not be suitable for VM consolidation in comparison to the T430 (due to having double the number of CPUs and half the N^* and N^+ values) for this specific problem size ($T_m = 100$ K). Figure 13 shows the speedups of the PMs on the VMs consolidated up to ten machines, for the five SUTs experimented.

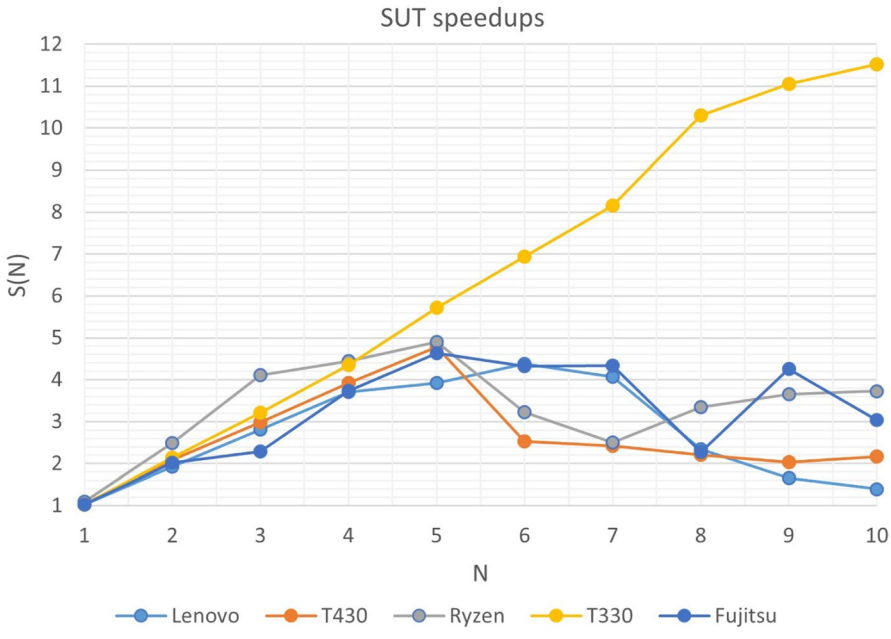

Fig. 13 Speedup of different PMs over consolidating of N VMs

Table 4 Speedup, efficiency by PM, efficiency VM, linearity and eficonsolidation ($T_m = 100$ K)

SUT	Dell T430	Dell T330	Ryzen 7	Lenovo ST550	Fujitsu RX500
#CPU	16	8	12	20	48
Processor	Intel Xeon E5-2600 v4	Intel Xeon E3-1200 v6	AMD Ryzen 7 5800X	Intel Xeon Platinum v2	Intel Xeon E7-4800 v3
RAM (GB)	8	16	32	128	1024
$S(N^*)$	4.7771	11.5259	4.9033	3.7081	4.6381
N^*	5	10	5	4	5
$E(N^*)$	0.9554	1.1525	0.9806	0.9270	0.9276
$\Theta°$ in N^*	43.6943	49.0548	44.4405	42.8320	42.8499
δ in N^*	0.0110	0.0018	0.0182	0.0109	0.0089
$S(N^+)$	2.5291	11.5239	2.4975	1.6604	2.2715
N^+	6	12	7	9	8
$E(N^+)$	0.4215	0.9603	0.3567	0.1844	0.2839
$EC(N^+)$	0.1044	0.0060	0.1055	0.0933	0.0932
$\Theta°$ in N^+	22.8569	43.8407	19.6358	10.4533	15.8518
δ in N^+	0.0458	0.0025	0.0492	0.0572	0.0430

Table 5 Speedup, efficiency, linearity and efficient consolidation

T_m	Dell T430				
	10K	50K	100K	200K	1000K
$S(N^*)$	4.4929	3.5138	4.7771	5.0974	6.8130
N^*	5	4	5	5	7
$E(N^*)$	0.8985	0.8784	0.9554	1.0194	0.9732
$E_{VM}(N^*)$	0.2225	0.2845	0.2093	0.1961	0.1467
$\Theta°$ in N^*	41.9422	41.2977	43.6943	45.5530	44.2246
δ in N^*	0.0153	0.0296	0.0110	0.0009	0.0029
$S(N^+)$	1.1990	2.5253	2.5291	2.6271	7.4779
N^+	2	6	6	8	10
$E(N^+)$	0.5995	0.4208	0.4215	0.3283	0.7477
$E_{VM}(N^+)$	0.8339	0.3959	0.3953	0.3806	0.1337
$EC(N^+)$	0.2288	0.1118	0.1044	0.0850	0.0269
$\Theta°$ *in* N^+	30.9446	22.8262	22.8569	18.1800	36.7890
δ in N^+	0.1940	0.0522	0.0458	0.0330	0.0050

8.3 Varying the size of the problem

When different problem sizes are considered, as shown in Fig. 5 in the problem statement section, the sensitivity to the problem workload size per machine becomes evident in N^* and N^+ (see Table 5). In brief, when a higher workload is distributed in independent tasks per machine, the parallel PMs gain efficiency over the consolidated VMs, and vice versa. Consequently, increasing the problem size results in a steeper linearity angle, denoted by $\Theta°$, while δ falls with the increase in N^* and N^+.

This was an expected result since the more computational requirements due to the size of the workload the less effective and efficient sharing one PM among N VMs.

Table 6 Mean execution times with different types of virtualizations

N	T_{PM}	$T_{VM\text{-}I}$	$T_{VM\text{-}II}$
1	24.2532	25.4260	26.2150
2	9.2707	19.2885	19.8390
3	5.2972	15.8380	16.1230
4	3.5626	13.9777	14.0680
5	2.6253	12.5415	13.5850
6	2.0386	5.1560	8.8390
7	1.5668	3.7863	–
8	1.3093	2.8964	–
9	1.1296	2.2928	–

8.4 Adding overhead

In the preceding experiments, a comprehensive approach was introduced to ascertain and assess the overhead in server consolidations, particularly concerning the consolidation of type I virtual machines. Table 6 provides an illustrative example of the diverse execution times as N varies, comparing type I and type II virtualizations for the Dell T430 server, which serves as our exemplary model. In the case of type II virtualization, the addition of a software layer further exacerbates the slowdown in mean execution times because CPU and RAM resources are more heavily utilized compared to type I virtualization. The type of virtualization does not hinder the application of the formulation from the earlier sections. The only consideration required is to factor in the new fixed overhead introduced by the operating system in OV_v, which increases the negative value of γ.

Table 6 shows that utilizing type II hypervisors in a time division distribution consolidation rapidly depletes resources due to the additional overhead imposed by the operating system. Consequently, deploying as many virtual machines as with type II virtualization becomes impractical, resulting in a reduced scope of N. Nonetheless, the entire formulation remains equally applicable (as shown in Table 7).

8.5 Redefining linear scalability views

In previous results, we have used efficiency, ($E(N)$), and the eficonsolidation index, ($EC(N)$), to determine the values of N^* and N^+. To facilitate a consistent and coherent way to visualize the speedup of N PMs over N VMs, as well as the performance and energy trade-off, the linearity constant has been also proposed. The following subsections offer more results than the examples offered in the development of the theoretical formulation.

In Figs. 14, 15, 16 and 17, we represent the examples of Table 5. As already anticipated, by varying the size of the problem, the N VMs consolidated in a PM can be a software parallelism option for relatively small independent similar task sizes. The values of the quotient of the hypotenuse of triangles $M(N)$ and N reveal the nonlinearity of the speedup $S(N)$ of N PMs over N VMs varying the size of the workload. The bigger the workload (T_m), the values are closer to the constant of linearity (red dotted line).

Table 7 Speedup, efficiency, $\Theta°$, $EC(N)$, γ and δ (Type II)

$S(N)$	$E(N)$	$\Theta°$	$EC(N)$	γ	δ
1.0808	1.0808	47.2260	0.0000	−0.0748	0.0000
2.1399	1.0699	46.9363	0.0054	”	0.0210
3.0436	1.0145	45.4141	0.0221	”	0.0233
3.9488	0.9872	44.6309	0.0234	”	0.0195
5.1746	1.0349	45.9833	0.0091	”	0.0136
4.3358	0.7226	35.8532	0.0597	”	0.0231

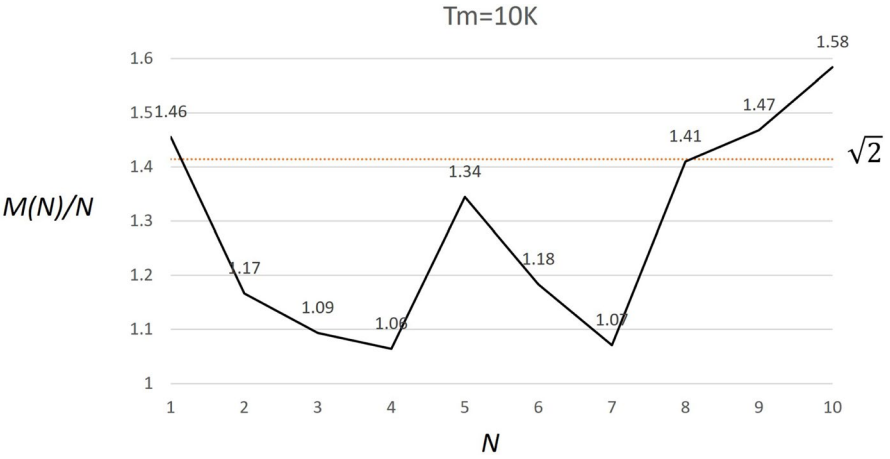

Fig. 14 Linearity of $S(N)$ over N for $T_m = 10$ K

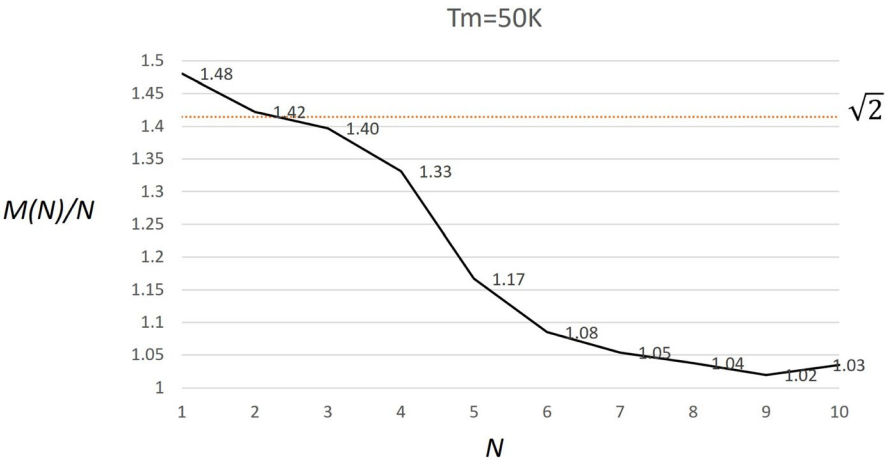

Fig. 15 Linearity of $S(N)$ over N for $T_m = 50$ K

8.6 Consistent selection of N^* and N^+

If we vary the problem size in the benchmark with the same example of the Dell T430 server, we see that there are small differences in the determination of N^* and N^+ compared to the results of the heuristic indicators in Table 5, specifically the values of $E(N)$ and $EC(N)$. In Figs. 18, 19, 20 and 21, we show the differences between consecutive hypotenuses $M(N)$ in the blue line, and the $M(N)/N$ ratio in the black line in comparison with the linearity constant (red dotted line). The algorithm selects the N^* and N^+ values, even though the graphical representation is clear enough to point them out when the size of T_m varies.

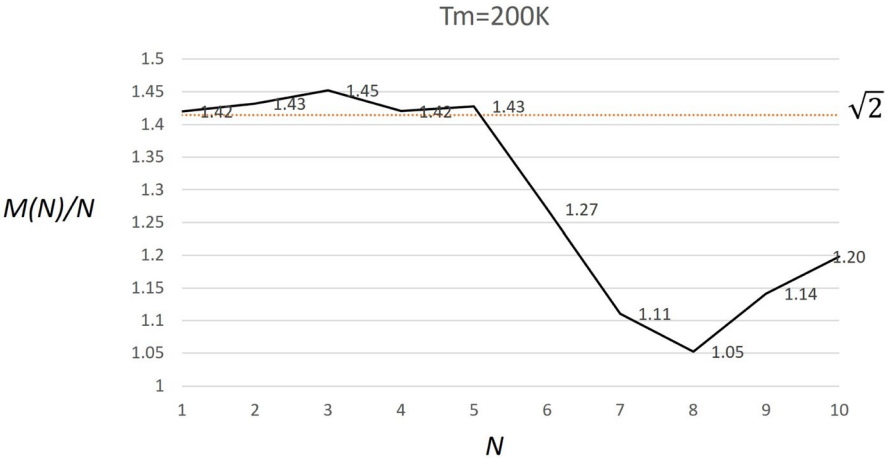

Fig. 16 Linearity of $S(N)$ over N for $T_m = 200$ K

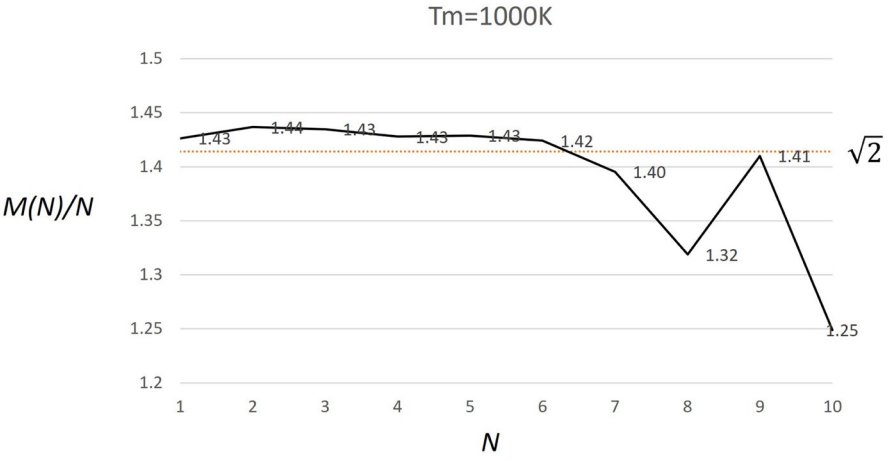

Fig. 17 Linearity of $S(N)$ over N for $T_m = 1000$ K

In Fig. 18, as we saw in Table 5, $N^+ = 2$, although we have alternatives for the selection of N^+ until $N \leq 4$. Also, now we can confirm that $N^* = 5$, seeing the difference of hypotenuses of 2.47, then drop the speedup of the N PMs over the N VMs to a sublinear slump, where it compensates for the consolidation overhead and this representation allows us to select $N^+ = 6$. Finally, we can see in Fig. 18 that for $N \geq 8$ the N PM accelerate superlinearly, due to the division into smaller tasks and the increase in the overhead of the N VMs. Figure 19 confirms that the comparison pattern with the linearity constant adjusts the choice of N^* and N^+ more consistently than the indicators that have been shown in previous sections. Thus, we can see that $N^* = 3$ and $N^+ = 5$, have shifted one machine, in comparison with the calculations

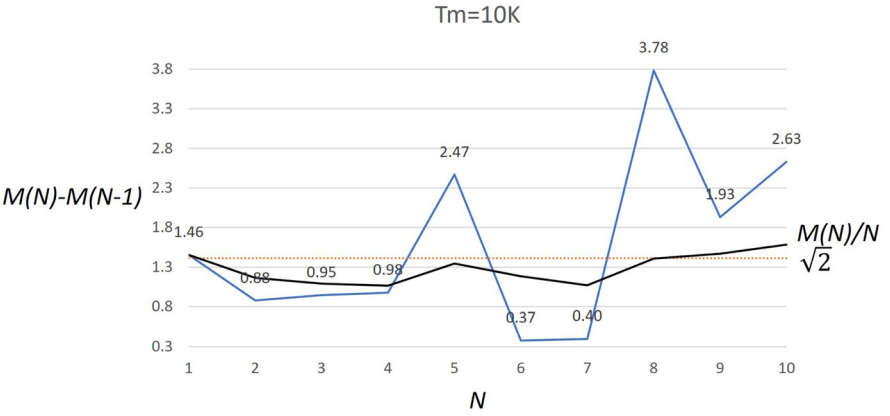

Fig. 18 Linearity and differences between neighboring consolidations for $T_m = 10$ K

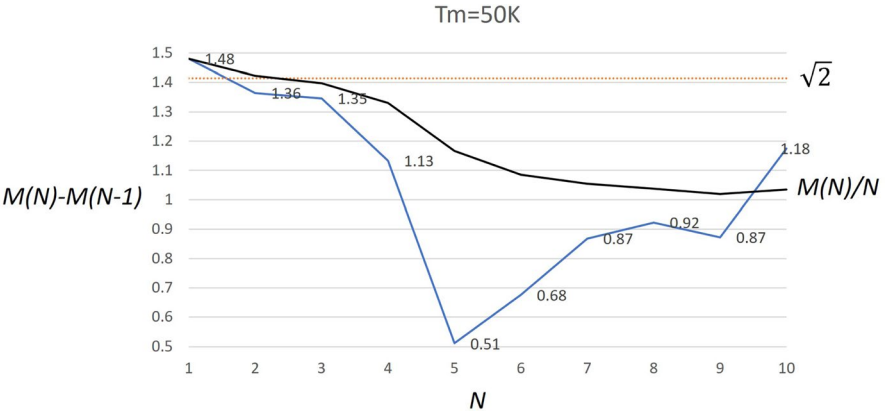

Fig. 19 Linearity and differences between neighboring consolidations for $T_m = 50$ K

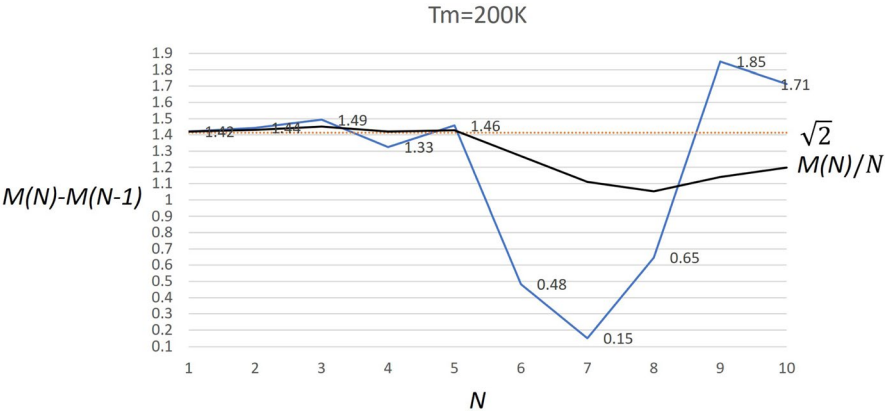

Fig. 20 Linearity and differences between neighboring consolidations for $T_m = 200$ K

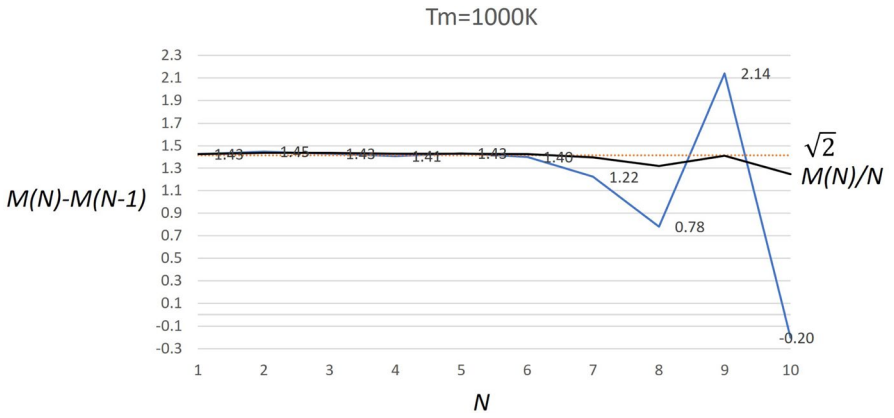

Fig. 21 Linearity and differences between neighboring consolidations for $T_m = 1000$ K

in Table 5. If the curve $M(N)/N$ is observed, the interval of maximum speedup of the N PMs on the N VMs with the smallest number of machines end up in $N^* = 3$.

However, in Table 5, $N^* = 4$, it has been decided that $E(4) = 0.87$ was sufficient. As demonstrated, the linearity constant pattern allows us to better fit $E(N^*) \approx 1$. On the other hand, in $N^+ = 5$ the greatest difference between hypotenuses is produced and not in $N^+ = 6$, where the indicator $EC(6)$ was the maximum (see Table 5). This is because the $EC(N)$ indicator performs an average among the N-considered machines and there may be a small displacement in the selection of N^+, concerning using the differential with the linearity constant. This is verified in the following examples of Figs. 20 and 21.

In Fig. 20, it is shown that $N^* = 5$ and $N^+ = 7$, while in Table 5, $N^+ = 8$ because $EC(8)$ is average with 8 machines. N^* matches Table 5, by the value of $E(5)$ compared to $E(6)$. In Fig. 18, it is confirmed that $N^* = 7$ (see Fig. 15 and Table 5), and $N^+ = 8$, although with two more machines $EC(10)$ is maximized and coincides with Table 5. Also, this example of Fig. 21, shows the risk of the sprawling phenomenon, due to the sawtooth that occurs when $N = 9$.

Table 8 illustrates the differences between N^* and N^+ when calculated using either $E(N)$ and $EC(N)$ or the linearity constant reference, i.e., the hypotenuse divided by N and the hypotenuse differences. The values presented in parentheses represent the second option for each method to choose N^+. By employing the linearity constant as

Table 8 Differences in the establishment of N^* and N^+

T_m	Dell T430				
	10K	50K	100K	200K	1000K
$N^*(E)$	5	4	5	5	7
N^*(linear)	5	3	5	5	7
N^+(EC)	2 (3)	6 (7)	6 (7)	8 (7)	10 (9)
N^+(linear)	2 (3)	5 (6)	6 (7)	7 (6)	8 (10)

a reference, we achieve a more unified and streamlined approach to determining N^* and N^+.

8.7 The good, the bad and the ugly

We have chosen the title of this subsection as a reference to a famous *spaghetti western* film directed by Sergio Leone and featuring a young Clint Eastwood, filmed in various locations in Spain [28].

Among the examples presented, we have selected Fig. 20, and colored it in three distinct areas, to illustrate the region recommended for parallelizing N PMs, the area recommended for consolidating N VMs, and the area where parallelizing tasks on a larger number of machines is not advisable in any case.

This visual representation allows for a clear and intuitive understanding of the consolidation options, making it easier to identify the most suitable scenarios for both parallelization and consolidation. The combination of efficiency indicators and the linearity constant ensures a comprehensive and well-balanced approach to server consolidation, contributing to the successful management of virtualized infrastructures.

Figure 22, likely illustrates the linearity of speedups for N PMs versus N VMs consolidated. By using the values of the ratios $M(N)/N$ and the differences of consecutive hypotenuses $M(N)$, we can determine specific zones in this representation. These zones likely correspond to regions where particular consolidation options are recommended or not recommended based on the speedup comparison. Based on the graphical representation provided earlier regarding the linearity constant ($\sqrt{2}$) and its relationship to the hypotenuses $M(N)$, it is possible to deduce the following:

- If the ratio $M(N)/N$ is greater than (or equal to) $\sqrt{2}$, it indicates that the speedup of N PMs over N VMs is increasing at a faster rate because the speedup is super-

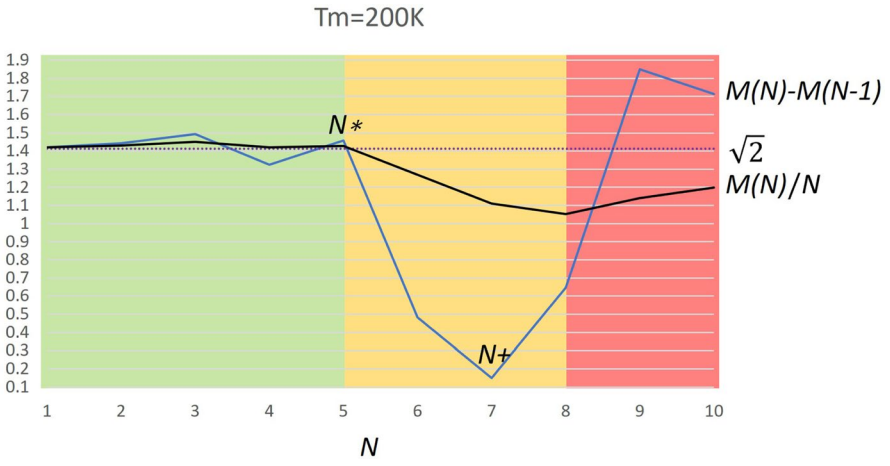

Fig. 22 Regions defined for server consolidation

linear (or linear). This zone might be favorable for parallelizing tasks on multiple machines (N PMs).
- If the ratio $M(N)/N$ is less than $\sqrt{2}$, indicates that the speedup of N PMs over N VMs is increasing at a slower rate. This zone might be more suitable for consolidating N VMs on one PM.
- The differences of consecutive hypotenuses $M(N)$ provide insights into the changes in speedup between different N values. A larger difference might indicate a more significant advantage for one approach over the other (parallelizing or consolidating). See the algorithm for the selection of N^* and N^+.

By analyzing these values and relationships in the graphical representation, one can effectively identify (from left to right):

- The "good" zone for PMs, represented in green, corresponds to the region where it is advantageous to parallelize N PMs, and this approach is limiting at N^*. The decision to parallelize or consolidate depends on the size of the workload to be executed (T_m). Typically, for most scenarios, maintaining N parallel PMs yields superlinear or linear speedups compared to consolidating N VMs. However, there are specific cases where the workload size is relatively small compared to the server resources (e.g., $T_m = 10$ K). In such cases, the "good" zone may exhibit sublinear speedups, with low $E(N)$, high $EC(N)$, and possibly containing some candidates for N^+. In these situations, it may be more favorable to consider consolidating N^+ VMs instead of parallelizing them on N PMs.
- The "bad" zone for PMs, shown in yellow, corresponds to the region where parallelizing N PMs becomes less advantageous and usually starts at $N^* + 1$. The decision to parallelize or consolidate in this zone also depends on the size of the workload to be executed (T_m). Here, N PMs accelerate sublinearly against N VMs, and N^* is still usually smaller than N^+. In this zone, it is often beneficial to consider consolidating N^+ parallel VMs, as most of the examples demonstrate. However, similar to the "good" zone, there are specific cases where the workload size is relatively small for the server resources (e.g., $T_m = 10$ K). In such instances, even though N^+ is smaller than N^*, some sublinear speedups may still be present in this "bad" zone, accompanied by low $E(N)$ values. To make well-informed decisions in this region, it is crucial to consider factors like speedup trends and efficiency indicators ($E(N)$ and $EC(N)$).
- The "ugly" zone for PMs and VMs, represented in red, is characterized by unfavorable conditions for both N PMs and N VMs. This zone starts when the N PMs exhibit again linear or even superlinear speedups compared to the consolidated N VMs, or on the contrary experience very low sublinear speedups. The benefits of parallelization diminish as the execution times of the tasks per machine become very small in proportion to the overhead involved. This leads to fluctuations in the linearity of the speedup, resembling a "sawtooth" pattern, where the performance alternates between superlinear and sublinear speedups. For N PMs, having a high number of machines in this zone would lead to increased space, cooling, and electrical power consumption in data centers. For N VMs, the "ugly" zone poses the risk of sprawling, where the

Table 9 Mean execution times (T) and power (W) of PM and VM type I

N	T_{PM}	W_{PM}	$T_{\text{VM-I}}$	$W_{\text{VM-I}}$
1	24.2532	94.2250	25.4260	111.3590
2	9.2707	188.8680	19.2885	111.8080
3	5.2972	283.3206	15.8380	112.6900
4	3.5626	365.3420	13.9777	111.1780
5	2.6253	438.5450	12.5415	111.2315
6	2.0386	508.2540	5.1560	108.9700
7	1.5668	583.3730	3.7863	104.8820
8	1.3093	658.4560	2.8964	104.4297
9	1.1296	734.3730	2.2928	97.7104

Table 10 Speedup, ratio of increased energy and CiS^2 of VM type I

N	$S(N)$	$S_e(N)$	$CiS^2(N)$
1	1.0483	1.2389	1.2988
2	2.0805	1.2316	2.5625
3	2.9898	1.1892	3.5555
4	3.9234	1.1939	4.6843
5	4.7771	1.2116	5.7882
6	2.5291	0.5422	1.3714
7	2.4165	0.4344	1.0498
8	2.2121	0.3508	0.7761
9	2.0297	0.2700	0.5481

consolidation becomes inefficient and resource utilization is suboptimal. The presence of the "ugly" zone is dependent on the number of samples (N) for a given workload size (T_m). In Fig. 19, only the first two zones may appear, but with a greater number of machines, the "ugly" zone would eventually emerge as shown in Fig. 22. This indicates that there is an optimal range for the number of machines, beyond which the efficiency of both N PMs and N VMs starts to decline.

As the size of the workload to be executed increases, the "good" zone (green) becomes wider, indicating that parallelizing N PMs remains a favorable option over consolidating N VMs. This wider "good" zone suggests that parallelization is efficient for a broader range of workloads and resource configurations. Similarly, the "bad" zone (yellow) appears later with larger sizes for workloads, indicating that consolidating N^+ VMs becomes disadvantageous compared to parallelizing N PMs for a broader range of workload sizes. The "ugly" zone (red) might not even appear or become less prominent with larger sizes for workloads and more machines, indicating that the inefficiency of both parallelization and consolidation is minimized in such scenarios. This aligns with the observation that more

machines cannot be consolidated for the specific server's resources, reducing the likelihood of the "ugly" zone's appearance, depending on the amount of hardware resources of the PM.

8.8 Consolidation and energy

CiS^2 is the product of the increased ratio in the energy of the consolidated N VMs over the N PM times the corresponding speedup. Table 9 shows the mean values of power consumed during the execution times of the tasks of a workload size $T_m = 100$ K, in the Dell T430 server, for the first nine machines. Table 10 shows the mean values of speedup, energy increase ratio and their product CiS^2.

Figure 23 shows the linearity of the $CiS^2(N)$ per machine, with the values of Table 10. As can be seen, the shape is practically the same as the linearity of $S(N)$ per machine in Fig. 8 but outlined by the values of energy ratio $S_e(N)$. Therefore, $S_e(N)$ acts as a weight in the speedup, increasing the superlinear or linear values, and, conversely, reducing the sublinear values. Looking at Fig. 23, we can confirm that $N^* = 5$ not only for performance but also for energy, since it is above the linearity pattern. In the same way, it is verified that the N VMs consolidated in a PM is an optimal option from $N \geq 6$.

Again, the difference between consecutive terms in a series involving the hypotenuses $M'(N)$ and $M'(N-1)$ is greater than $\sqrt{2}$, meaning that the terms are increasing at a faster rate. That is the difference between $CiS^2(N)$ and $CiS^2(N-1)$, which is what changes in the calculation of the hypotenuses, changes to faster performance and energy consumption of the N PMs over the consolidated N VMs. Conversely, if the difference is less than $\sqrt{2}$, the terms increase at a slower rate.

In Fig. 24 we have shown both the linearity constant and the $M'(N)/N$ curve of Fig. 23, to also represent the differences between consecutive M' hypotenuses. It can be observed that, indeed, the period where the N PMs accelerate

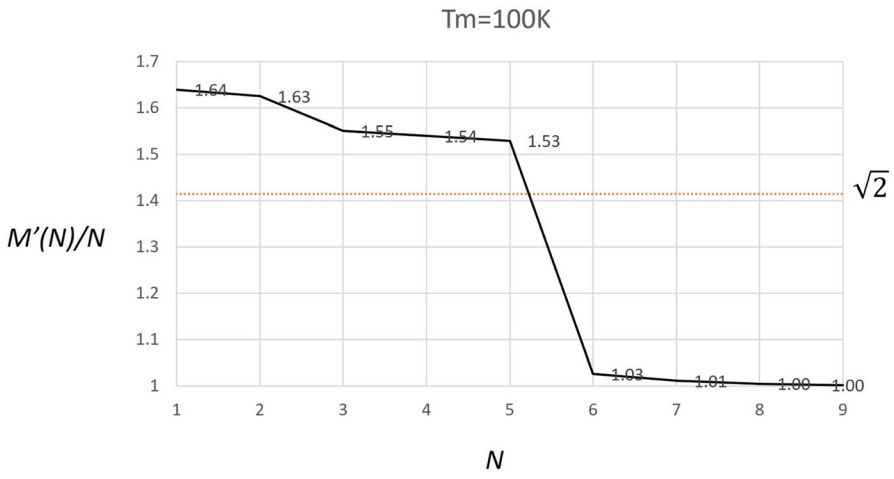

Fig. 23 Linearity of the performance and energy trade-off of N PMs over N VMs for $T_m = 100$ K

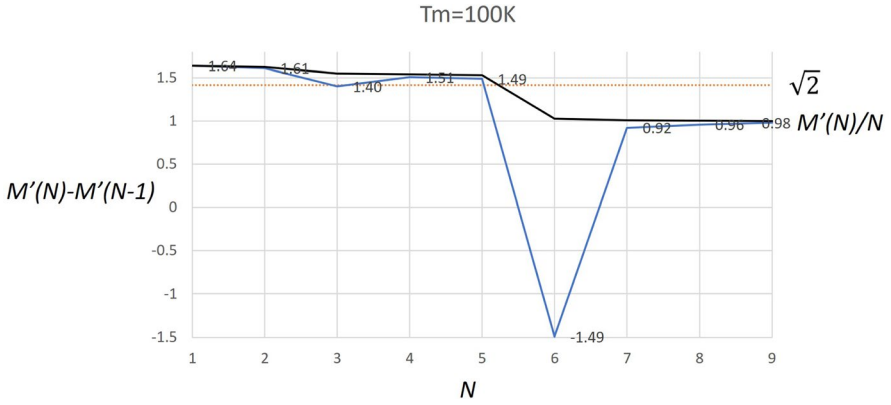

Fig. 24 Linearity of $CiS^2(N)$ over N and differences between neighboring consolidations for $T_m = 100$ K

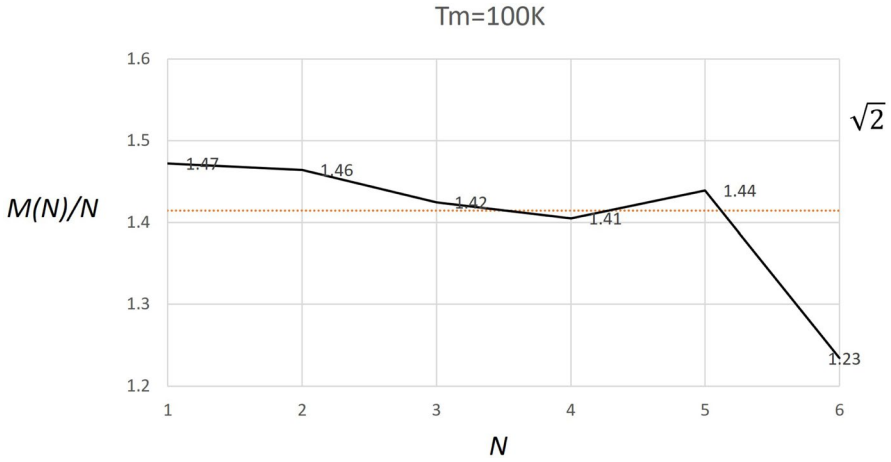

Fig. 25 Linearity of $S(N)$ over N for $T_m = 100$ K (type II)

its performance and increase the energy consumption linearly or above it, ends in $N^* = 5$. We also check that $N^+ = 6$, since it is the maximum negative difference of the hypotenuses with a value of negative 1.49. The demonstration that the energy only outlines the speedup linearity curve is the similarity between Figs. 10 and 24. In the case of using consolidation of type II virtual machines, the linearity constant similarly reflects the same characteristics as in type I. Figures 25, 26 and 27 show the same example of $T_m = 100$ K, but with the values of type II consolidation, shown in Tables 6 and 7.

As can be seen in Fig. 25, while $N \leq 5$ the N PMs, CiS^2 accelerate linearly, and in Fig. 25 it can be seen that $N^* = 5$ and $N^+ = 6$, the same as in the example of virtualization type I. The same values are selected with the linearity of the performance and energy relationship, as shown in Fig. 26.

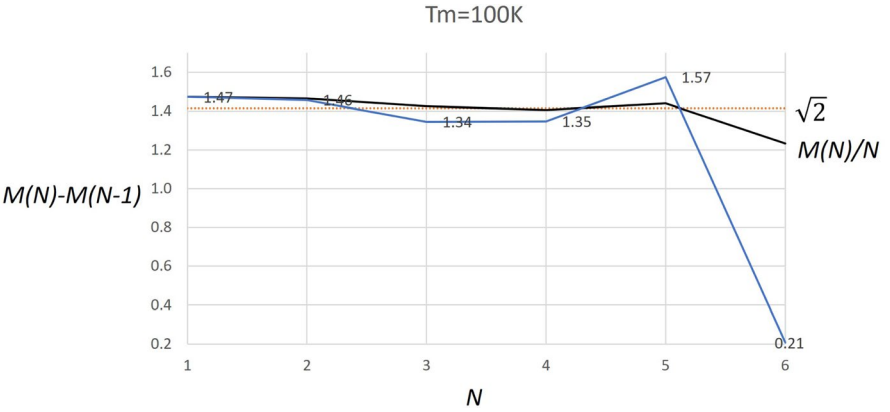

Fig. 26 Linearity and differences between neighboring consolidations for $T_m = 100$ K (type II)

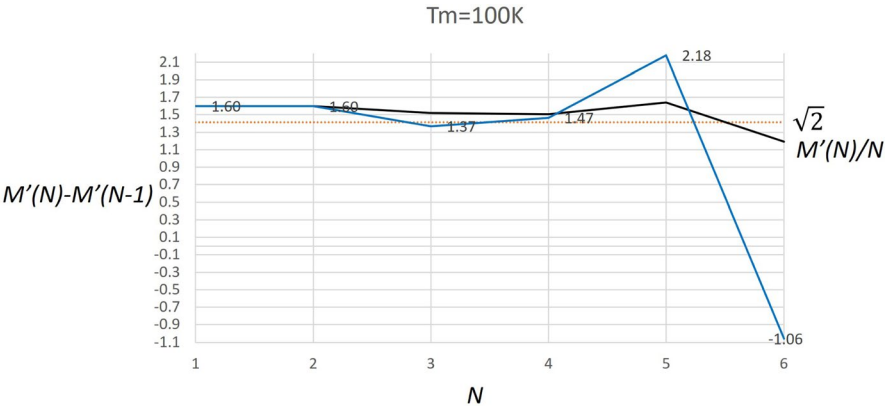

Fig. 27 Linearity of $CiS^2(N)$ over N and differences between neighboring consolidations for $T_m = 100$ K (type II)

Regarding the differences in virtualization type, when N VMs are consolidated, type I have higher superlinear performance and energy than type II, before N^*, and lower sublinear performance and energy after N^*. In N^*, it happens that the performance and energy of type II is higher than type I. Figure 28 demonstrates that adding the operating system in the virtualization of type II causes a constant added overhead, in comparison with type I virtualization, not using a layer of additional software. This only pays off in performance and energy for N^*, that is when it is better not to consolidate N^* PM.

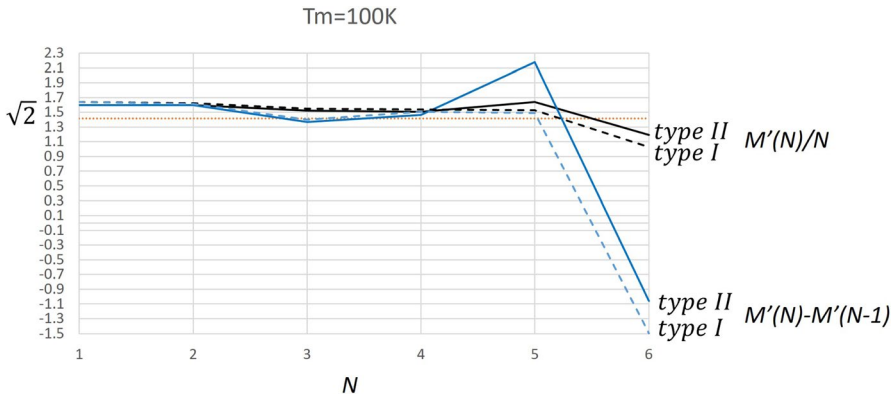

Fig. 28 Linearity of $CiS^2(N)$ over N and differences between neighboring consolidations, virtualization type I and type II

9 Discussion

This research has been focused on formalizing speedup and efficiencies used in parallelism, to determine the nonlinear behavior of speedup for consolidation of virtual machines in comparison with the physical machines' parallelization of independent tasks. We also provide heuristics and a consistent graphical method to determine the optimal degree of consolidation for virtual servers, using intensive CPU workloads and to a lesser extent RAM. No other types of workloads, hypervisors or benchmarks have been taken into account for the experimentation than those mentioned. However, other experiments through other CPU-intensive workload benchmarks would produce results quantitatively different for different workloads and overheads, although the formulation should be fully applicable.

Since the study is based on the speedups of a machine against itself, with the same workload, the same hypervisor and the same conditions, the comparison is fair. Naturally, considering mean execution times, each hardware performs differently depending on its resources, especially CPU, for a given size workload, i.e., the mean execution times are different depending on resources. Servers with few resources tolerate virtualization well if the workload is small, and vice versa, servers with more resources tolerate overheads better, for a given problem size. Precisely, the workload is distributed evenly in parallel independent tasks, but it could have been scaled in concurrent multitasks since the result would be analogous as shown in [19].

This paper concentrates on the problem of how to formalize speedup and overheads modifying classical formulations comparing PMs against VMs and proposing new ways to represent scaling for CPU-intensive workloads. We demonstrated in previous publications [5, 23, 29] and experiments with different transactions but CPU intensive that even though the magnitudes are different, the physical machines and virtual machines behave similarly.

All the research has been carried out with homogeneous servers, to establish the degree of virtualization comparably, not only between different options of the same PM but also between different PMs. The heterogeneity of machines has been studied in previous works, as well as their nesting, for example, using containers inside consolidated VMs in PM as described in [6], but using mean execution times and not speedups or efficiencies, as in this research work.

In data center scenarios, it is crucial to carefully manage resource allocation and ensure that PMs are not overloaded, which could lead to performance degradation and energy inefficiencies. Balancing workload distribution and resource allocation becomes critical to maintaining the advantages of virtualization and consolidation while avoiding potential pitfalls associated with resource contention.

Additionally, adopting power management strategies, such as dynamic resource allocation and load balancing, can help optimize energy consumption and maintain the benefits of virtualization even under high utilization conditions. By dynamically adjusting resource allocation based on workload demands and PM utilization, data centers can strike a balance between performance, energy efficiency, and cost savings.

Overall, virtualization and consolidation offer substantial benefits in data centers, but effective management and optimization are essential to maintain these advantages under varying workload conditions and resource utilization levels. Strategic planning and monitoring of resource usage will contribute to a more efficient and cost effective virtualized environment.

But dynamic balancing, VM migration, resource allocation and other management and optimization actions are not under the scope of this research, since we should be seeking how much virtualization supports a particular server with a CPU workload in comparison to not using consolidation. In that context, it is essential to assess the trade-offs between virtualization and consolidation without considering dynamic management actions such as VM migration, resource allocation, or load balancing. The research concentrates on identifying the optimal number of PMs (named N^*) and the maximum number of VMs (named N^+) that can be efficiently consolidated on a specific server with a given workload. By determining N^* and N^+, we can establish the range within which virtualization offers the most significant benefits in terms of resource utilization and performance improvement. Additionally, comparing the results to the nonconsolidated scenario provides valuable insights into the efficiency gains and cost savings achieved through virtualization. By focusing on N^* , and N^+ , and the comparison with nonconsolidated scenarios, our research will provide valuable insights into the efficiency and benefits of virtualization in supporting specific servers and workloads. It will contribute to the understanding of how virtualization can be effectively utilized to optimize resource usage and improve performance in data centers, complementary to the potential need for dynamic management and optimization actions.

9.1 Theoretical implications

The use of the Pythagorean constant ($\sqrt{2}$) as a reference for establishing linear scalability in virtualization may have significant potential impact and benefits in

the field of performance evaluation and scalability analysis. Some of the potential impacts and benefits of this research work include:

- Improved graphical representation: The use of the Pythagorean constant in graphical representations provides another clear and intuitive way to visualize and understand the linearity of scalability. This approach can complement the visualization of performance and resource utilization trends, making it easier for researchers and practitioners to interpret the results.
- Simplified explanation of scalability: The Pythagorean constant allows for straightforward explanations of scalability phenomena, not only in performance and energy aspects but also in various other scenarios.
- Consistent framework for scalability analysis: By using the Pythagorean constant as a reference framework, the research establishes a consistent and coherent approach to evaluate scalability across different studies. This can lead to more standardized and comparable results in the field of performance evaluation and scalability analysis.
- Broad applicability: The concept of using a reference constant for scalability analysis can have applications beyond virtualization and data centers. It can be adapted and extended to other domains that require scalability assessment, such as distributed systems, cloud computing, and large-scale computing infrastructures.

This research work may have the potential to influence the way linear scalability phenomena are explained and evaluated, not only in the context of performance and energy, nor for consolidation only, but across diverse fields that require a clear reference framework for scalability analysis.

In the example of Fig. 29, it can be seen how the speedup would be transformed with the Amdahl, Gustafson and Gunther (USL) laws, using as a vertical axe the quotient between the hypotenuse and the number of machines, $M(N)/N$, instead of

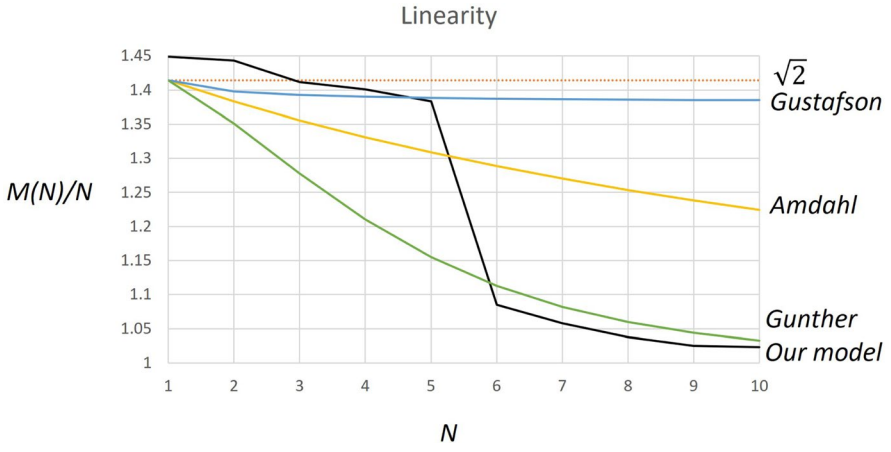

Fig. 29 Linearity representativeness of different speedup laws (example)

using the classic $S(N)$. It seems easy to compare the linearity of the three examples against the constant of linearity (Pythagorean constant). We also represent (black line) the particular example which corresponds to the values of consolidation in Fig. 8 (δ is not a constant) so that the consolidation curve has a superlinear phase and a sublinear one.

The three laws applied to the example are taking N as the degree of concurrency, the contention fraction $\alpha = -\gamma$ has been selected for Amdahl and Gustafson, and the fixed coherence fraction $\beta = \overline{\delta}$ or Gunther's USL law, to illustrate another way of speedup linearity representativeness.

The findings developed in this research can be extended and applied to a wide range of virtualization technologies and scenarios beyond the specific context in which the research was conducted. Some key aspects that contribute to the generalizability of the study include:

- Versatility in scalability analysis: the Pythagorean constant approach and the determination of N^* and N^+ can be applied to any software-parallelizable machine, regardless of the specific virtualization technology used.
- Wide range of problem sizes: the research's methodology is not limited to a specific size of CPU-intensive problems. As long as the workload can be divided into equal tasks or is scalable in parallel multiple tasks, the scalability analysis using the Pythagorean constant can be effectively applied.
- Adaptability to future virtualization technologies: the research's generalizable nature makes it well-suited for application to future virtualization technologies that may emerge.

9.2 Practical implications

There may be tangible consequences of this research for practitioners, especially system and data center administrators. Here are some specific benefits for practitioners:

- Scalability assessment: by calculating the speedup and comparing it to the linearity constant (Pythagorean constant), administrators can determine the level of scalability achieved in the virtualized servers. The distance from the linearity constant indicates how well the virtualized system scales and where it lies on the efficiency spectrum.
- Optimal resource utilization: the research's insights into N^* and N^+ provide administrators with guidelines on the optimal number of virtual machines to be consolidated on a particular server for efficient resource utilization. This helps in avoiding resource overutilization or underutilization, leading to cost savings and better performance.
- Energy and performance trade-off: armed with the knowledge of the consolidation index (CiS^2) and scalability metrics, administrators can make informed decisions regarding workload management, virtual machine consolidation, and resource allocation. This ensures that data centers run in an optimized and cost effective manner.

- Problem positioning: the research enables administrators to identify specific areas, such as the "good," "bad," or "ugly" zones, where virtualization may excel or perform sub-optimally. This allows them to address any performance issues, prevent sprawl, and optimize the virtualized environment.

System administrators can still effectively utilize the presented models of speedup calculations. For example, executing real tasks as benchmarking comparison with different configurations adding or subtracting a VM sharing the workload accordingly [30]. The proposed models can be used for baseline comparisons between different system configurations, architectures or technologies, allowing to identification of relative performance improvements or energy efficiencies. Therefore, system administrators can conduct sensitivity analyses to assess the robustness of their servers to variations in several configuration factors. This can help in identifying critical parameters and understanding the boundaries within which configurations remain valid. By considering various site-specific factors in different scenarios, system designers can assess the potential impact of changes in machine characteristics, workload profiles, user behavior, and resource availability on the expected speedup.

This research is part of a project that intends to propose the extension of the ISO/IEC 30134 [23] standard, which contemplates only the performance and energy efficiency of physical servers without considering virtualization. The study conducted here may lead to technological advancement and a direct impact on the server industry by suggesting how to standardize the measurement of consolidation performance and energy.

9.3 Limitations and future work

Virtualization technology drives the consolidation of VMs or containers. Despite their similar functionality, there are significant differences between them in terms of performance (as measured by mean execution time), security, implementation, and portability. These differences affect consolidation decisions when choosing between virtual machines or containers, and their degree of consolidation. Traditionally, servers are consolidated by assigning virtual machines or containers to a physical server. However, mixing virtual machines and containers on the same physical server can mitigate the drawbacks of both. In a previous work, the authors proposed a method to quantify the magnitude of the consolidation overhead time from the average execution time of a task in the PM, comparing it with a number (N), either of VMs, or containers. Subsequently, the server consolidation overhead time estimation method was generalized for any combination (configuration) of arbitrary consolidation of VMs and containers, regardless of their configuration or nesting. It is one of the future works of this research, to delve into the nesting [29] of containers and consolidation.

All results of this research are considered for CPU-intensive workloads for servers and only take into account the execution time (service time). No queueing or latency is considered. Although it is obviously outside the scope of this work, in a case study carried out with virtual machines in a real company, the analysis has

been extended with types of transactional CPU and memory workload, whose performance has been measured with other metrics based on queuing networks [7]. That is, taking into account the mean waiting time and the mean response time of the transactions to execute the corresponding tasks. The results do not contradict, but complement what is presented here about mean execution tasks as mean service times.

Our proposed method could be adapted to emerging trends in virtualization and cloud computing, mainly on containerization that provides a lightweight and portable approach to application deployment, contributing to the agility and scalability of cloud environments.

Our paper concentrates on the problem of how to formalize speedup and overheads modifying classical computer architecture formulations and proposing new ways to represent scaling for CPU-intensive workloads, using a black-box model. That is, servers or virtual servers running tasks and meanwhile measuring their execution time and consumed power. This addresses the paper?s fundamental questions of formalization and representativeness and not underlying the causes of the scalability shapes. It is part of our future work to investigate further the relationship between the lower-level performance metrics and the changes in scalability due to the reduction of the size of the problem and the augmented overhead due to the incremental fractionality of the workload.

10 Conclusions

This work proposes the use of basic concepts of parallelism, such as speedup, efficiency, isoefficiency and scalability, adapting them to quantify and represent the overheads of consolidation of virtual machines in physical machines and study the balanced relationship between performance and energy when consolidating virtual machines. The proposed formalization has been extensively experimented with different physical machines and different sizes of tasks to verify the theoretical approach. An attempt has also been made to represent, first intuitively, by heuristics, the distance from the ideal of linear parallelism, when software tools are used to parallelize machines, to later determine a comparative method for speedup and balance between performance and energy based on elementary trigonometry. The results of this research make it possible to establish how far to parallelize a physical server and how far to consolidate virtual machines optimally, but adaptable to the circumstances that system managers support in data centers. In summary, the generalizability of the research work extends its applicability to a wide range of virtualization technologies and computing scenarios. It offers a versatile and adaptable framework for scalability analysis and performance evaluation, making it valuable not only for current virtualization technologies but also for future developments in the field of software-parallelizable computing. The potential cross-domain applications highlight the relevance of the research's findings in optimizing resource utilization and performance across diverse computing environments.

This research work may have the potential to influence the way linear scalability phenomena are graphically explained and evaluated, not only in the context of

performance and energy, nor for consolidation only, but across diverse fields that require a clear reference framework for scalability issues.

Author contributions All the authors contributed equally in this paper.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work is part of the Project TED2021-132695B-I00, funded by MCIN / AEI / 10.13039 / 501100011033 and by the European Union "NextGenerationEU" / PRTR. We thank the University of Seville (Spain) and the Hasso-Plattner Institute (Germany), for the use of their SUTs in different R &D projects. Finally, we want to pay tribute to Hippasus of Metapontum, who is believed to have discovered that the square root of two was an irrational number.

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethical approval Not applicable.

References

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin A, Stoica I, et al (2009) Above the clouds: a Berkeley view of cloud computing. Technical report, technical report UCB/EECS-2009-28, EECS Department, University of California
2. Wang W, Chen H, Chen X (2012). In: 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing (IEEE), pp 509–516
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. Lindner M, McDonald F, McLarnon B, Robinson P (2011). In: 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops (IEEE), pp 1062–1065
5. Bermejo B, Juiz C (2021) On the classification and quantification of server consolidation overheads. J Supercomput 77(1):23–43
6. Bermejo B, Juiz C (2022) A general method for evaluating the overhead when consolidating servers: performance degradation in virtual machines and containers. J Supercomput 78(9):11345–11372

7. Juiz C, Capo B, Bermejo B, Fernández-Montes A, Fernández-Cerero D (2023) A case study of transactional workload running in virtual machines: the performance evaluation of a flight seats availability service. IEEE Access 11:81600–81612. https://doi.org/10.1109/ACCESS.2023.3300956
8. Dias AH, Correia LH, Malheiros N (2021) A systematic literature review on virtual machine consolidation. ACM Comput. Surv. (CSUR) 54(8):1–38
9. Singh J, Walia NK (2023) A comprehensive review of cloud computing virtual machine consolidation. IEEE Access 11:106190–106209. https://doi.org/10.1109/ACCESS.2023.3314613
10. Songara N, Jain MK (2023) Mra-vc: multiple resources aware virtual machine consolidation using particle swarm optimization. Int J Inf Technol 15(2):697–710
11. Zolfaghari R, Sahafi A, Rahmani AM, Rezaei R (2021) Application of virtual machine consolidation in cloud computing systems. Sustain Comput Inform Syst 30:100524
12. Huber N, von Quast M, Hauck M, Kounev S (2011) Evaluating and modeling virtualization performance overhead for cloud environments. CLOSER 11:563–573
13. Hwang K, Jotwani N (1993) Advanced computer architecture: parallelism, scalability, programmability, vol 199. McGraw-Hill, New York
14. Al-hayanni MAN, Xia F, Rafiev A, Romanovsky A, Shafik R, Yakovlev A (2020) Amdahl's law in the context of heterogeneous many-core systems-a survey (2020). IET Comput Digit Tech 14(4):133–148
15. Hennessy JL, Patterson DA (2011) Computer architecture: a quantitative approach. Elsevier, Amsterdam, pp 36–55
16. Amdahl GM (1967). In: Proceedings of the April 18–20, 1967, Spring Joint Computer Conference, pp 483–485
17. Gustafson JL (1990) In: Proceedings of the Fifth Distributed Memory Computing Conference (DMCC5). IEEE Press, pp 1255–1260
18. Shi Y (1996) Reevaluating Amdahl's law and Qustafson's law. http://www.cis.temple.edu/~shi/docs/amdahl/amdahl.html
19. Gunther NJ (2006) Guerrilla capacity planning: a tactical approach to planning for highly scalable applications and services. Springer, Berlin
20. Juiz C, Bermejo B (2020) The c i s 2: a new metric for performance and energy trade-off in consolidated servers. Clust Comput 23(4):2769–2788
21. Conway JH, Guy R (1998) The book of numbers. Springer, Berlin
22. Gonzalez R, Horowitz M (1996) Energy dissipation in general purpose microprocessors. IEEE J Solid-State Circuits 31(9):1277–1284
23. I (2020) 30134-5, ISO/IEC 30134-4:2017 information technology data centres key performance indicators
24. Casalicchio E (2019) A study on performance measures for auto-scaling CPU-intensive containerized applications. Clust Comput 22(3):995–1006
25. Buyya R, Vecchiola C, Selvi ST (2013) Mastering cloud computing: foundations and applications programming. Newnes, Oxford
26. Jain R (1991) The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling, vol 1. Wiley, New York
27. Kounev S, Lange KD, von Kistowski J, Kounev S, Lange KD, Kistowski Jv (2020) The SPEC CPU benchmark suite. In: Systems Benchmarking: For Scientists and Engineers, pp 231–250
28. Caprara V (2006) Il buono, il brutto, il cattivo: storie della storia del cinema italiano. Guida Editori, Naples
29. Chae M, Lee H, Lee K (2019) A performance comparison of Linux containers and virtual machines using Docker and KVM. Clust Comput 22(Suppl 1):1765–1775
30. Desai PR (2016) A survey of performance comparison between virtual machines and containers. Int J Comput Sci Eng 4(7):55–59