J Grid Computing
https://doi.org/10.1007/s10723-018-9440-8

# A Cost Model for IaaS Clouds Based on Virtual Machine Energy Consumption

**Mauro Hinz · Guilherme Piegas Koslovski · Charles C. Miers · Laércio L. Pilla · Maurício A. Pillon**

Received: 14 November 2016 / Accepted: 24 April 2018


**Abstract** Cloud Computing has revolutionized the software, platform and infrastructure provisioning. Infrastructure-as-a-Service (IaaS) providers offer on-demand and configurable Virtual Machine (VMs) to tenants of cloud computing services. A key consolidation force that widespread IaaS deployment is the use of pay-as-you-go and pay-as-you-use cost models. In these models, a service price can be composed of two dimensions: the individual consumption, and a proportional value charged for service maintenance. A common practice for public providers is to dilute both capital and operational costs on predefined pricing sheets. In this context, we propose PSVE (Proportional-Shared Virtual Energy), a cost model for IaaS providers based on CPU energy consumption. Aligned with traditional commodity prices, PSVE is composed of two key elements: an individualized cost accounted from CPU usage of VMs (e.g., processing and networking), and a shared cost from common hypervisor management operations, proportionally distributed among VMs.

**Keywords** IaaS · Virtual machines · Cost · Model · Energy

M. Hinz
Brazil, IBM, Porto Alegre, RS, Brazil
e-mail: mmhinz@gmail.com

G. P. Koslovski (✉) · C. C. Miers · M. A. Pillon
Graduate Program in Applied Computing, Santa Catarina State University, Joinville, SC, Brazil
e-mail: guilherme.koslovski@udesc.br

C. C. Miers
e-mail: charles.miers@udesc.br

M. A. Pillon
e-mail: mauricio.pillon@udesc.br

L. L. Pilla
Department of Computer Science, Federal University of Santa Catarina, Florianópolis, SC, Brazil
e-mail: laercio.pilla@ufsc.br

## 1 Introduction

The demand for data processing, storage and communication has been motivating the development of scalable, on-demand and ubiquitous Information Technology (IT) platforms over recent decades. It is notorious the pervasiveness of IT on professional and personal daily tasks. In this context, the way computing resources are offered and delivered to customers is under continuous evolution [36, 39]. Currently, data storage and processing are hosted by specialized Data Centers (DCs), geographically distributed and accessible by Internet, or in other words, IT services are hosted in a cloud computing environment [32]. Among all services delivered by clouds, the Infrastructure-as-a-Service (IaaS) providers offer on-demand and configurable VMs to tenants. VMs are provisioned atop virtualized DCs, designed to host multiple isolated instances. Indeed, IaaS is recognized as a popular technology commonly acquired for

Published online: 28 May 2018

migrating services from local DCs to large-scale cloud providers [17, 43]. The adoption of IaaS was facilitated as VMs are regularly used for over-provisioned server consolidation.

A key consolidation force that popularized the IaaS deployment is the use of *pay-as-you-go* and *pay-as-you-use* cost models. These utility-oriented charging models are not new, as they are traditionally used in the supply of basic resources such as water and electricity [44]. Following these models, IaaS clouds tenants are no longer responsible for acquiring and maintaining dedicated hardware and now they just reserve a set of VMs for a predetermined time frame, or while needed. However, on traditional commodities, a service price is composed of two dimensions: one representing the individual consumption, and a second one charging a proportional value for service maintenance [8]. For instance, energy delivery companies are allowed by regulatory agencies to proportionally charge street light, while water distribution operators can charge ecological fees. In short, the service maintenance price is proportionally charged among using tenants. Although IaaS providers aim to deliver virtual resources as a utility commodity, the cost model is an open question, as private costs of capital expenditure and operational management must be considered while composing the price sheet.

A common practice for public providers is to dilute both capital and operational costs on predefined pricing sheets. For instance, tenants of Amazon EC2 can select a VM from a set of instance flavors without knowledge of internal provider investment on hardware and management. This approach is understandable as providers have no incentive to open internal data related with cost composition. *However, we claim that energy-aware cost models are an opportunity for improving this scenario.*

Recent studies highlighted that energy consumption is a major cost for DCs [13, 42]. Moreover, almost 2% of world-energy consumption is due to DCs [12, 21, 25]. Despite appealing social aspects, the reduction of energy consumption stands out for its immediate economic impact on IaaS providers [30, 38, 46]. Data center electricity consumption is usually organized into two main groups: consumption of the general DC infrastructure, and consumption directly related to the equipment associated to supply the computing resources (e.g., servers, storage devices, networking equipment) [3, 13]. Specifically, CPU servers figure out as a major fraction of the consumption [10, 13, 14, 39, 40]. For decreasing the energy consumption, some techniques have been applied to migrate VMs balancing the DC load [22, 24, 37], and to deactivate idle cores, processors or servers [7, 15].

Looking closely to the charging model of a VM instance on public cloud providers, it can be noted that the pricing does not change according to the submitted processor workload. Specifically, Amazon EC2 adopts a criteria for charging the provided VM instances based on the time of the instance usage (active) and its configuration (the flavor, e.g., number of vCPUs, amount of storage, data traffic). *As the processor of a computer is a major consumer of electricity, the opportunity to take into account the electricity consumption of the VM instance on the charging model is clear.* Moreover, traditional charging models are not aligned with proportional cost sharing among tenants. In order to illustrate, lets consider two VMs with equivalent configuration and distinct CPU workload: 10% for $vm_a$ and 90% for $vm_b$. While $vm_a$ represents a web-server with idle working periods, $vm_b$ represents a CPU-intensive application. For a single hosting server, $vm_a$ and $vm_b$ have distinct impacts on CPU energy consumption but both are usually charged the same price. Hence, it does not matter for the providers if a VM instance is using 5% or 100% of its processors, because the value charged remains constant.

Virtualization brings a new opportunity for energy-based cost sharing: as a hypervisor has full knowledge of hosted VMs loads, the energy consumption can be quantified per VM. Indeed, a naïve approach may just quantify virtual CPU (vCPU) usage and its real load on physical devices. However, such accounting is incomplete: independent of virtualization technology, a subset of VMs calls has impact on the hypervisor's processing load. For example, network input and output calls increase CPU usage, as VM requests are multiplexed on hardware. In short, any VM operation that reflects on physical CPU usage must be quantified and individually charged. This discussion illustrates our second claim: *as hypervisors have full accounting knowledge on virtual and physical CPU usage, a proportional energy-cost sharing can be developed.*

In this context, this work proposes PSVE (Proportional-Shared Virtual Energy), a cost model for IaaS providers based on CPU energy consumption. Aligned with traditional commodity prices, PSVE is composed of two key elements: an individualized cost accounted from CPU usage of VMs (e.g., processing and networking), and a shared cost from common hypervisor management operations, proportionally distributed among VMs.

The rest of this paper is organized as follows: Section 2 describes the motivation and related work. Section 3 accounts two relationships between VM workload and energy consumption, serving as base for composing PSVE. The first aspect quantified is the electricity consumption induced by different processor loads. Afterwards, CPU consumption originated from virtual networking interfaces is quantified. PSVE is described in Section 4, while a discussion faced to Amazon EC2's cost model is presented in Section 5. Section 6 concludes the work presenting future perspectives.

## 2 Motivation and Related Work

Before introducing a cost model based on VM energy consumption, a discussion on the charging models currently applied by public IaaS cloud providers is required. After this discussion, related work is reviewed identifying academic and commercial proposals related with IaaS cost and pricing models, and energy consumption.

### 2.1 Current VM Charging Approaches of IaaS Cloud Providers

Although cloud computing displays certain features, such as automatic provisioning capabilities, one of its main objectives is to take advantage of economies of scale by an optimized resource usage [45]. Achieving an optimal pricing for the service is a challenge for cloud service providers. Providing a single efficient pricing model for all different models of cloud computing (IaaS, PaaS, and SaaS) is really complex. In [23], the pricing model is classified as: *Pay-as-Use*, *Subscription-Based*, and *Hybrid pricing model*. However, in this work, we consider only IaaS cloud providers and their specific characteristics. IaaS cloud providers use cost models that combine multiple metrics and rules to set the price for the use of their resources [8]. The cloud environment, policies, rules, or requirements are oriented to the business model in which the computing resources (e.g., processing, storage and networking) are services that are measured as energy and water commodities. This model is known as *utility computing* and usually is implemented using other computing infrastructures based on additional policies and rules for accounting and monitoring services [16]. However, a standardization of these policies and rules was not identified on the specialized literature. The degree of automation, abstraction and customization defining a service can be considerably different. There are providers offering to their users the possibility of simple construction conditions based on certain metrics, such as: number of cores (vCPU), amount of memory, storage, and network usage. Nevertheless, other providers may adopt metrics in the service level (e.g., cost-benefit ratio) and allow even more complex strategies [49] such as composing VMs based on specific application requirements (e.g., memory slice and variable vCPU usage). In short, cloud computing presents two main charging models:

- *Pay-as-you-go*: the client pays for the allocated resources. The client will pay a fixed value for resources, whether using them or not [35]; and
- *Pay-as-you-use*: the client pays only for the resources really consumed. Computer resources are like a service provided automatically as needed and charged according to their consumption [6].

These models, or collection strategies, may change in granularity and measurements adopted from provider to provider. In order to identify which models are the most widely adopted, Table 1 compares the cost models of some of the main IaaS providers according to the Gartner Group IaaS Report [28].

Besides technological choices (e.g., virtualization type, VM components, hypervisor), the comparison in Table 1 considers a set of characteristics related to cost model composition, such as: *pay-as-you-go*; *pay-as-you-use*; energy consumption charged by usage; hypervisor influence over cost; type of network charges; and internal/external traffic charges. Table 1 also shows that the listed providers do not consider in their cost models the processor usage/workload, only the number of processors allocated per VM instance. However, a survey analyzed several midsized data centers (5,000-square-foot) in the USA in order to reveal how the energy consumption is usually distributed, pointing out a relevant amount of energy consumption per processors [14].

**Table 1** Aspects considered in the charging process of main commercial IaaS providers

| Provider | Amazon EC2 | IBM | Microsoft Azure | Rack space | Google Compute Engine |
|---|---|---|---|---|---|
| Characteristic | VM charging aspects | | | | |
| *Pay-as-you-use* | Yes | Yes | Yes | Yes | Yes |
| *Pay-as-you-go* | Yes | Yes | No | Yes | Yes |
| Energy consumption charged by usage | No | No | No | No | No |
| Hypervisor influences costs | No | No | No | No | No |
| Differential charging by DC location | Yes | No | Yes | No | Yes |
| Characteristic | VM Aspects | | | | |
| Virtualization type | VM, Container | VM, Container | VM, Container | VM | VM, Container |
| VM components | CPU, ECU, MEM, DISK | CPU, MEM, DISK | CPU, MEM, DISK | CPU, MEM, DISK | CPU, MEM, DISK |
| VM charging unit | Instance type | Instance type | Instance type | Instance type | Instance type |
| Hypervisor adopted | Xen Server | Xen Server | MS-Azure hypervisor | Xen Server | KVM |
| Characteristic | Charging network usage | | | | |
| Type of network charges | GB | GB | GB | GB | GB |
| Internal traffic | No | No | Yes | No | No |
| External traffic | Yes | Yes | Yes | Yes | Yes |

Figure 1 identifies this energy consumption distribution, informing the proportions of the total energy used by different components, revealing that most of the energy consumption of a DC is related to the IT equipment, with a special portion of it being related to processor usage. Moreover, according to said survey,

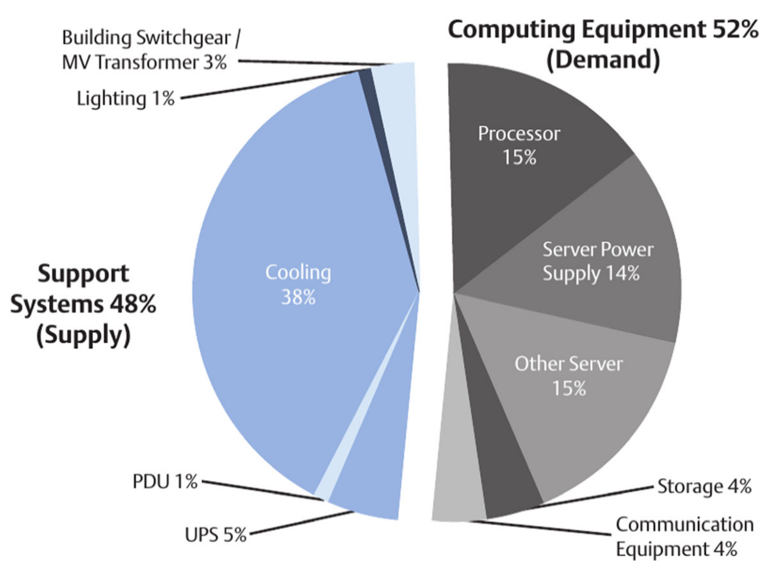



**Fig. 1** Energy consumption sources of a midsize data center [14]. Computing equipment consumption is 52% and processor consumption is the highlight of this group with 15%

the consumption of each Watt by the processor implies on the consumption or savings of nearly 2.84 Watts on the DC facility [14]. This cascade effect is explained in Fig. 2.

*We argue that the cost models aforementioned are more "pay-as-you-go" than "pay-as-you-use" because they do not consider the usage/workload of the processor (and its impact on the energy consumption of the DC) in the formation cost of the VM instance service provided. Thus, the workload of a processor on a server (hosted by an IaaS cloud computing provider) is usually assigned to a VM. Consequently, we must verify the amount of energy consumed by a VM vs. the processor workload in order to state how relevant the latter is to the cost model.* Defining such relationship is a requirement for the proposal of a cost model based on virtual machine energy consumption, as discussed in Section 3.

## 2.2 Related Work

Identifying constant and variable components of a pricing model is a critical step for the correctness of any cost model. For IaaS cloud providers, a cost model must be aligned with costs of capital expenditure and operational management related with service provisioning. In this context, the specialized literature comprehends administrative and management perspectives, as well as energy consumption and sharing.

Gmach et al. [18] propose three cost models based on CPU workload, each one exploring a specific metric to infer the processing cost from the workload. An initial proposal calculates the cost from the average CPU consumption perceived for a given workload allocation. An alternative proposal for cost definition is to account the difference between the maximum and average CPU usage for a given application. Finally, a third proposal quantifies the cost based on the idle resources. All three proposals have individual advantages according to usage scenario, and can be applied to quantify the provisioning cost of an IaaS provider. However, all proposals lack energy efficiency and energy-aware costs accounting.

The ratio of CPU and storage energy consumption under different workloads was investigated by Shekhar et al. [41]. The authors concluded that the main cause of inefficiency of energy management in DCs is occasioned by long periods of server inactivity. Even for a low workload (e.g., CPU $\leq 10\%$), a server's energy consumption is greater than 50% of its peak utilization. Following this approach, if the disk, network, or any Input/Output (I/O) device is the application's performance bottleneck, the energy consumption is equivalent to idle CPU servers. Complementary, Lee

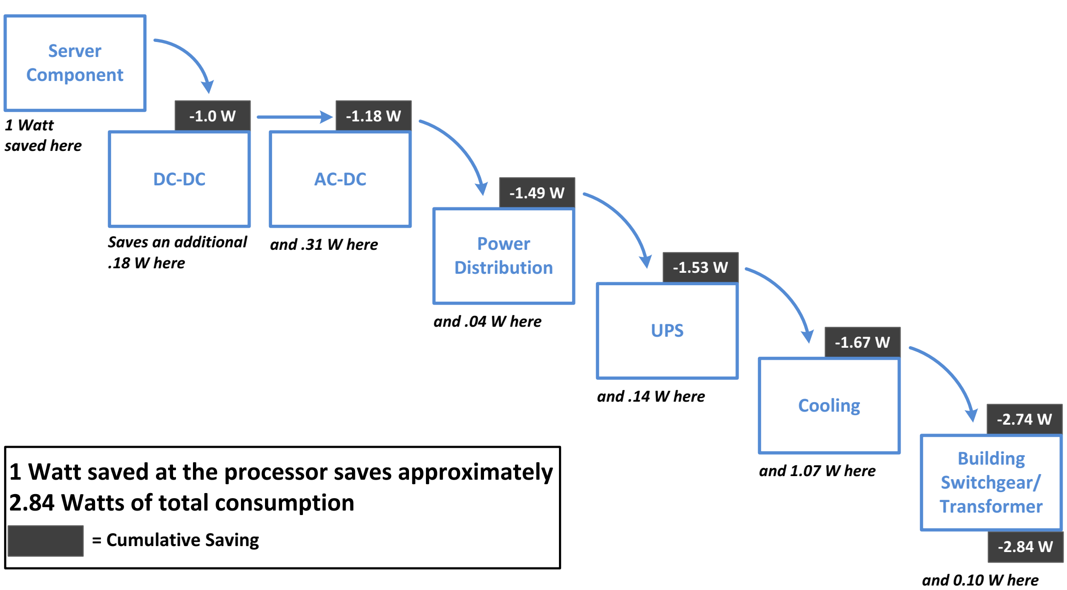



**Fig. 2** Cascade effect: saving 1 Watt at the server-component level creates a reduction in facility energy consumption of $\approx 2.84$ Watts [14]

and Zomaya [26] proposed a cost model based on CPU workload but it is applicable on non-virtualized scenarios only.

Server consolidation and resource sharing are well-know benefits of virtualization techniques. Moreover, the virtualization layers appeared as an opportunity to eliminate the gap between energy consumption sharing and accounting between hosted VMs. In this sense, the individual accounting of VM's resources and the related energy consumption was investigated on specialized literature [33, 47]. This research approach motivated the investigation of the relationship between energy consumption and processor workload performed in Section 3. The cost model proposed in the present work is based on this energy consumption and workload ratio.

By reviewing energy-aware management literature, a set of common requirements was identified for the composition of a cost model for IaaS cloud providers: *(i)* virtualization techniques are widely applied on DCs; *(ii)* a precise ratio between energy consumption and VM (virtual CPU) usage must be defined; *(iii)* a pricing model can be developed from an energy-aware cost model; *(iv)* providers must be able to identify which energy costs are related with service provisioning [19, 27]; and *(v)* I/O operations performed by VMs consume energy and, consequently, must be individually accounted.

The first characteristic specifies if the model is virtualization-aware, enabling a correlation between application workload and virtual CPU usage. Usually, workload-based cost models make no differentiation, considering traditional resources only [18, 26, 41]. The ratio identified in requirement *(ii)* was roughly accounted in some studies [26, 41].

The pricing model for cloud computing has been discussed by the academic community [2, 29]. Aldossary et al. [2] defined a cost model based on operating production factors, production, sales theory, investment, and finance. This model considers the relationship between energy consumption and target load, but does not take into account the individual and collective costs according to VM resource usage. In cloud computing environments, a fair cost model should consider the VM consumption for each individual tenant. Mach and colleagues [29] defined a cost model based on VM resource usage. This simple model adds energy cost to individual price of VM according to consumption by individual VM resources (e.g., CPU, memory, disc, and network). However, the energy consumption identification of an individual VM resource did not take into account the operations performed by the hypervisor only (e.g., VM scheduling, I/O drivers). Thus, if a VM sends a message to another, this operation is performed by the hypervisor and, consequently, CPU usage is counted in the hypervisor account. In this case, energy consumption for this shipping message has not been added to the VM account.

Usually, the literature lacks mechanisms that consider items (iii) and (iv) together. None of the analyzed models consider the requirements in their cost models. Thus, this fact shows a gap in the conception of cost models suitable for sharing energy costs on IaaS providers. Finally, although I/O devices have a smaller impact on energy consumption, a cost model that primes for proportional division of energy-related costs should consider at least the use of network operations (requirement *(v)*). Naturally, the use of remote data storage on IaaS clouds is a common technique (e.g., Amazon S3 services / OpenStack Glance, Cinder, and Swift). This approach increases network usage and consequently the related energy consumption. The cost model proposed in Section 4 complies to the requirements, performing a fair distribution of energy consumption between hosted tenants.

## 3 Analysis of Energy Consumption and CPU Utilization Rate

For processing the workload of a VM-hosted application, the virtualization hypervisor must schedule and address an available vCPU to the target VM. Thus, any application's instruction submitted to a vCPU is processed by the physical processor [9]. It is a fact that such operations are energy consuming and can be correlated with VM CPU utilization rates. Indeed, the higher the CPU usage, the greater is the energy consumption [26]. However, a set of hypervisor calls related with VMs' management and networking operations is not directly translated to CPU utilization and requires further investigation.

For composing a cost model based on CPU energy consumption, all CPU consuming aspects must be taken into account. Besides accounting the VM individual consumption, a proportional sharing of hypervisor consumption among hosted VMs must be performed (depicted in Fig. 3).

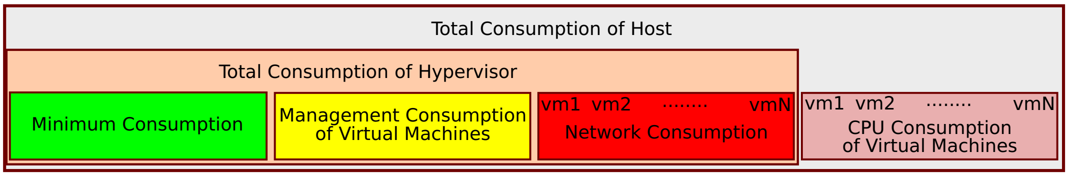



**Fig. 3** Components related on the energy consumption in virtualized environments. Three of four components are managed by the hypervisor and the energy consumption is counted within hypervisor consumption. The only exception is the CPU energy consumption that is linked to a VM when one vCPU is assigned

Consequently, as an investigation of baselines is needed to delineate our cost model, we focus on three aspects: minimum CPU energy consumption; correlation of energy and CPU utilization; and network traffic energy consumption.

### 3.1 Experimental Scenario and Methodology

In order to identify the energy consumption proportion of each component of VMs, we define a scenario representing a private IaaS provider. We use this testbed because we need to have total control of the workload submitted to the VM instances, and we need to install the wattmeter hardware for a precise energy consumption measurement. In this scenario, a standard rack server is directly attached to the wattmeter and then to the energy grid (non-intrusively), while a regular desktop equipment is used to collect and process energy consumption data. The testbed configuration is detailed as follows:

- *Host Server*: Model 6027R-TRF, two Intel Xeon E5-2620 2.0 GHz (12 cores total, 24 threads due to HT), 64 GB (DDR3) of RAM, two 1 TB SATA disks (7200RPM), 740 W redundant power supply, and four Intel i350 Gigabit Ethernet RJ45 network interfaces. Hypervisor software: XenServer 6.2SP1. VM instances: GNU/Linux CentOS 6.6.
- *Desktop*: HP Compaq model 6005, one AMD FX-6300 3.5 GHz (6 cores), 16 GB (DDR3) of RAM, one 1 TB SATA disk (7200RPM), 320 W standard power supply, and one Realtek 8111E Gigabit Ethernet network interface. Operating system: GNU/Linux CentOS 6.6.
- Switch Dlink DGS-3100-24, 24 ports (10BASE-T/100BASE-TX/1000BASE-T).
- Wattmeter: Model Wats Up? PRO, 220V/60Hz/15amps, capacity to store in local memory 1700 record samples, USB interface for real-time reading, and accuracy of the electricity consumption reading ±1.5%.

We adopted the following software for workload generation and monitoring:

- *stress*[1]: used to create benchmarking workload on processor and memory.
- *iperf*[2]: used for packet injection (both TCP and UDP) and for measuring the network bandwidth usage.
- *cpulimit*[3]: benchmarking tool to limit the CPU usage of a given process.
- *CPU load script*[4]: controls the workload generated by *stress* and *cpulimit*. This shell script creates configurable workloads.
- *XenServer metrics*[5]: tool for monitoring the XenServer hypervisor. The data is collected through HTTP by standard SOAP requests.
- *Monitor*: a tool for collecting data, used for calibration and monitoring. The shell script feeds a database with energy consumption data from the VM instance utilization by vCPU, processor usage rate, and amount of network data usage per VM.

The experiments are divided into three test groups:

1. Minimum and maximum energy consumption: the initial calibration aims to identify the CPU energy consumption limits. The minimum consumption measurement occurs when the host is turned on with just its basic services running (e.g.,

[1] http://people.seas.harvard.edu/~apw/stress/
[2] http://sourceforge.net/projects/iperf/
[3] http://cpulimit.sourceforge.net/
[4] https://github.com/ajurge/CPU_load
[5] http://xenserver.org/partners/developing-products-for-xenserver/18-sdk-development/96-xs-dev-rrds.html

hypervisor) and no active VMs, while the maximum represents the consumption with all CPUs saturated.
2. Energy consumption *vs.* CPU utilization rate: we used the *CPU load script* to generated workloads in a range from 5 to 100% of processor usage. Data collection is done through the monitoring script.
3. Energy consumption *vs.* networking operations: by generating traffic from and to VMs, the energy consumption related to CPU hypervisor operations is accounted by the monitoring script.

For each scenario, 10 executions were performed, and each datapoint corresponds to a 5-minute average (data is measured and collected every second). Before accounting the data, an initial calibration (2 minutes) is performed in order to stabilize the CPU workload, avoiding unwanted variations of the start-up process. The CPU workload scenario goes from 5% to 100% using 5% increments, while workload for network-related tests are injected from 100 Mbps to physical limit (1000 and 11000 Mbps for one and two hosts, respectively).

### 3.2 Minimum and Maximum Energy Consumption

The minimum and maximum energy consumption are components that must be calibrated for each host platform. This calibration is justified by consumption differences between architectures, and by the impact of the versions of the virtualized operating system and hypervisor. The obtained power limits measured for the target host are 116.88 W and 202.43 W, so the minimum consumption is 58% of the peak consumption.

### 3.3 Energy Consumption *vs.* CPU Utilization Rate

This experiment consists on applying fixed workloads to a server to correlate the CPU workload with its effective energy consumption. Moreover, one can suggest that a workload may have different energy consumption when applied to a virtualized server, even when using all available vCPU. In this sense, this test case is decomposed into two others:

– Test 1: workload applied on the hypervisor; and
– Test 2: workload applied on a VM instance with 24 vCPUs.

Figures 4 and 5 summarize the results: the horizontal axis is the processor workload while the vertical axis is the power consumption in watts. Each line represents the number of vCPUs being used in the test. When applying the workload on the hypervisor (Fig. 4), the effective use of a CPU is representative for accounting the energy consumption: even applying the lighter workload (5%), an increment of ±15% was observed over the minimum baseline with 24 vCPUs.

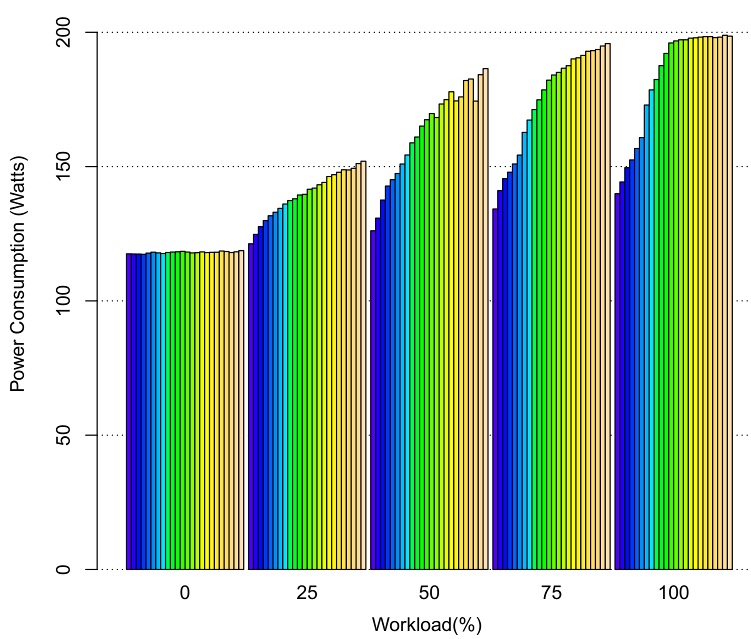



**Fig. 4** Test 1: Energy consumption *vs.* CPU usage, workload applied on the hypervisor. The vCPUs (1 to 24) are in the $x$-axis, starting by blue bars and ending by beige bars. Naturally, the energy consumption for all CPUs is similar when the workload is at 0%. When the workload is 100%, we can observe an increase in energy consumption according to the CPU number

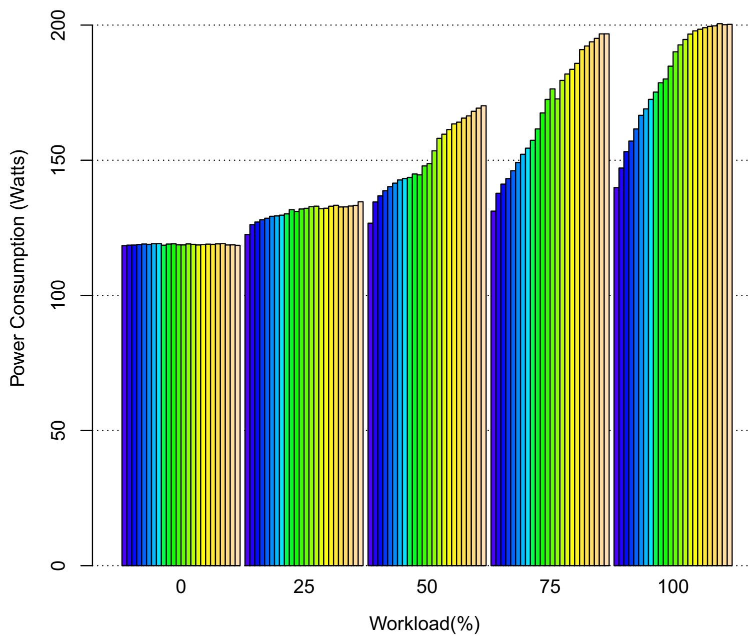



**Fig. 5** Test 2: Energy consumption *vs.* CPU usage, workload applied on a VM instance. The vCPUs (1 to 24) are in the $x$-axis, starting by blue bars and ending by beige bars. Naturally, the energy consumption for all vCPUs is similar when the workload is at 0%. When the workload is 100%, we can observe an increase in energy consumption according to the CPU number

In order to facilitate the reading, Fig. 4 presents only some workload results (0%, 25%, 50%, 75%, 100%). However, we measured data with a 5% periodicity, as needed for composing the discussion in Section 5. In general, the profile of Test 1 (Fig. 4) has a linear trend with two well-defined ranges: initially between 10% and 45%, and in a second moment between 55% and 100%. However, both ranges have distinct growing functions. In the first range (10% and 45%), the consumption grow follows the load increment, even with points where a smaller number of vCPUs consumed more than other with more vCPUs allocated. The second range (55% and 100%) has a less significant growth rate as it approaches the physical cores saturation. For this scenario, the highest standard deviation (2.29%) was identified at 188.99 W with a workload of 60% and 24 vCPUs.

Regarding the workload variation on the VM instance (Test 2), Fig. 5 indicates that the results follow the previous scenario. However, the workload seems to have a greater influence on consumption than the number of active cores. In this scenario, a 15% increment on energy consumption was only achieved with 25% workload, against 5% for Test 1. A less severe gradual growth can be observed with workloads between 0% and 30%. After 35%, there is a sharp growth until the core saturation point (between 95% and 100%). For this scenario, the highest standard deviation (5.35%) was observed at 184.79 W when the workload was 100% with 12 vCPUs.

*The obtained results (Tests 1 and 2) allow us to state the impact of the processor usage on the electricity consumption. Thus, the use of the "pay-as-you-use" model to charge a VM instance active using 5% of the processor the same value charged to the another VM (identical hardware/software configuration) using 95% of the processor does not seem to be befitting with the idea of a "pay-as-you-use" model. As a matter of fact, it is possible to estimate the energy consumption of a given server based on effective CPU utilization.*

### 3.4 Energy Consumption *vs.* Networking Operations

In IaaS clouds, VMs are usually provided with virtualized network interfaces. As the number of virtual interfaces can differ from the physical one, the hypervisor must handle hardware access sharing and multiplexing. Moreover, according to virtualization technology used by the provider, the hypervisor is the sole entity in charge of managing network traffic [4]. In this sense, the network I/O traffic of VMs impacts on the hypervisor's CPU utilization, even when source and destination are hosted by the same physical server [48]. Specifically, recent proposals have presented packet inspection and update for improving the congestion control of DCs [11, 20],

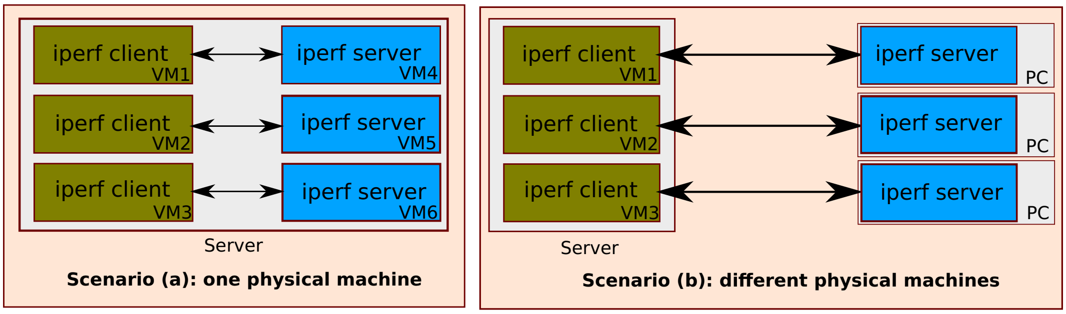



**Fig. 6** Test scenarios to evaluate the relationship between throughput and hypervisor processor usage. While scenario (**a**) explores the virtual network limit, in which the client and the server are on the same machine, scenario (**b**) targets the observation of client and server behavior on different machines

increasing the CPU energy consumption originated by the hypervisor. In brief, the energy consumption for data transfer (generated by VMs) is represented by the processor energy consumption for processing and forwarding packets.

In order to quantify the energy consumption related with VM networking, scenarios (a) and (b) from Fig. 6 were defined. The experiment relies on client-server pairs of *iperf* for generating network load. For scenario (a), clients and servers are hosted on a single virtualized server, while for scenario (b) a desktop machine was used for hosting the *iperf* servers. In both scenarios, clients inject constant network loads between 100 Mbps and the physical limit (11000 Mbps for scenario (a) and 1000 Mbps for scenario (b)). For all tests, the hypervisor was provisioned with a single vCPU.

Figures 7 and 8 present the results for scenarios (a) and (b), respectively. The $x$ axis represents the workload in logarithmic scale while the $y$ axis indicates the CPU utilization by the hypervisor to perform the packet transfer. Each point represents the average of 30 executions. In order to represent a shared scenario, up to three pairs of client-server were deployed, hosted by individual VMs.

Initially, Fig. 7 indicates that the hypervisor's CPU utilization increases with effective VM throughput. Using three client-server pairs, the CPU utilization with 100 Mbps is 2.95%, scaling up to 74.25% with a 11000 Mbps throughput. It is worthwhile to mention that such throughput is achieved as packet forwarding is performed internally on the hypervisor, without using physical network interfaces. Meanwhile, Fig. 8 shows that the impact on CPU utilization is lower as the effective throughput is limited by network interface capacity. For all tests in this scenario, CPU utilization ranges from 2.75% to 15.18%. As expected, the highest consumption is observed near to the physical interface saturation.

*By analyzing results from scenarios (a) and (b), we can observe that CPU utilization originated from network data transfer cannot be overlooked. Even when*

**Fig. 7** Scenario (**a**): CPU utilization for different throughputs using a single server to host all VMs. As the aggregated virtual interface throughput increases, more hypervisor processor is demanded. In the worst case, virtual network management caused consumption of more than 74% of hypervisor processor

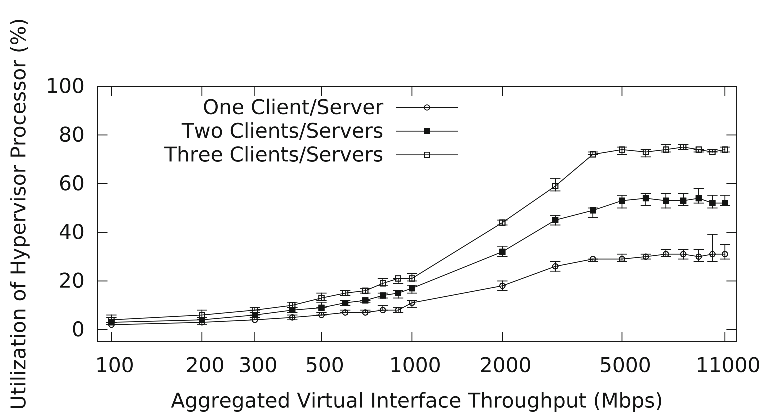
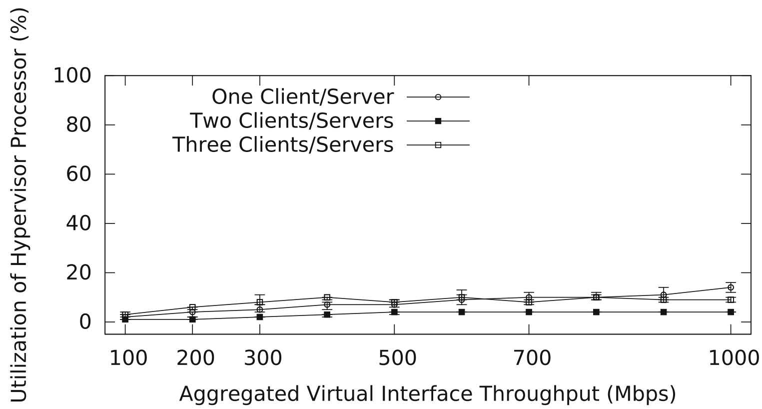


**Fig. 8** Scenario (**b**): CPU utilization for different throughputs with VMs executing *iperf* clients on the same server, while VMs executing *iperf* servers are hosted on another machine




*the communicating VMs are hosted by a single server, the CPU utilization required for packet transfer scales up to* 70%. *Translating such CPU utilization to energy consumption (Fig. 4), we can observe that server consumption is increased by* 20% *from its minimum value. Moreover, the energy consumption from CPU usage on a shared host is related to the number of hosted VMs.*

### 3.5 Discussion and Key Observations

Based on the sharing and isolation premises for virtualized infrastructures [34], the energy consumption of a vCPU, mapped atop a physical processor, has no impact on the energy consumption of other physical processors. Nonetheless, management tasks (scheduling and multiplexing) performed by a hypervisor have a considerable energy consumption which is not directly attributed to hosted VMs. In fact, management tasks are independent of VM CPU workload. In this sense, to schedule a vCPU with 0% or 100% CPU utilization has the same impact on hypervisor energy consumption, independently on VM. *The CPU utilization on VMs does not impact the hypervisor CPU load, but the opposite occurs with resource management.*

Moreover, the hypervisor processing to schedule four vCPUs to a single VM or to four distinct VMs is the same, and so is the energy consumption. The energy consumption originated by a given vCPU is uniquely dependent on its own workload. *Therefore, we can conclude that the energy consumption of a virtualized server is impacted by the number of active vCPUs, regardless of the number of vCPUs associated to VMs.*

On virtualized environments, the existence of active and idle VMs without workload is common. In this scenario, the individualization of energy consumption per VM is essential to identify the effective provisioning cost. Scheduling and multiplexing resources is performed on-demand by the hypervisor, i.e., this management entity associates a physical processor if and only if the VM requires it. *An idle VM has no processing demand, and consequently, no vCPU is associated.*

Following the experimental analysis, Fig. 3 summarizes the energy consumption components on a virtualized host: (i) minimum energy consumption: hypervisor energy consumption without active VMs; (ii) management energy consumption: referent to hypervisor processing to schedule, allocate and multiplex resources; (iii) networking energy consumption: related with hypervisor processing to handle VM data transfers; and (iv) VM energy consumption: originated by VM workload.

Finally, we can conclude that the energy consumption of two vCPUs, with $x$ and $y$ utilization rates, is equivalent to a single vCPU energy consumption with $w$ utilization rate, if and only if $w = x + y$. Thus, the proportionality can be generalized as indicated by (1), where $M$ and $N$ represent sets of vCPUs and $l_x$ represents the utilization rate (load) of vCPU $x$.

$$M \equiv N \iff \sum_{m \in M} l_m = \sum_{n \in N} l_n \tag{1}$$

The experimental analysis, key observations, and theoretical virtualization premises motivate the development of a cost model based on proportional sharing of energy consumption for IaaS cloud providers. Certainly, the model must consider all components detailed in Fig. 3, as discussed in the next section.

## 4 PSVE – Proportional-Shared Virtual Energy

In order to define the price of a given service commodity, a set of costs must be considered by a provider, such as: cooling, operational expenditure, and energy consumption. PSVE is a cost model for Infrastructure-as-a-Service (IaaS) cloud computing based on the energy consumption of the processors allocated to VMs. The model accounts a cost for a single VM, and following the analysis discussed in Section 3, ponders the energy-consuming components (individual and collective). It is worthwhile to highlight that although PSVE is not intend to address pricing models, a discussion on integration possibilities is addressed on Section 4.3 to support the comparison performed with current providers (Section 5).

Table 2 resumes the notations used to express PSVE. For this model, we consider a system with a set of VMs $\mathcal{V}$, a set of hosts $\mathcal{H}$, and a set of network interfaces of VMs $\mathcal{N}$. Besides relations returning subsets of the aforementioned sets, the model includes functions that can return CPU usage (load, in %) as $L$, bandwidth (in Mbps) as $B$, power (in watts) as $W$, energy (in Joules) as $E$, and a number of vCPUs as $C \subseteq \mathbb{N}$. Additionally, some functions consider a time period of analysis defined by the IaaS provider $T$ ($[t_{start}, t_{end}], t_{start}, t_{end} \in \mathcal{T} \subseteq \mathbb{R}_{\geq 0}$) or a discrete time instant $t$ $\{t \in \mathcal{T} \mid t_{start} \leq t \leq t_{end}\}$.

### 4.1 Identifying the Unit for Energy Consumption Proportionality

For performing a proportional collective consumption sharing, a ratio unit must be defined. Considering that CPU use is the host's largest energy consumer, the usage of a virtual CPU (vCPU) as the unit to divide collective consumption is an appropriate option to strive for proportionality in resource sharing among the soliciting users. Figure 9 illustrates a set of vCPUs (different number per VM) sharing some physical CPUs. The physical resource is multiplexed between vCPUs by the hypervisor scheduler. For instance, vm-1 has four vCPUs provisioned, however, only two are scheduled atop physical processors (physical CPUs 3 and 4 are temporary allocated to vCPUs 3 and 4). In this example, vCPUs 1 and 2 of vm-1 are not consuming energy, while vCPUs 3 and 4 are active and processing.

**Table 2** Notation used to represent PSVE

| Function notation | Description |
|---|---|
| $vCPU_h : \mathcal{H} \rightarrow C$ | vCPUs in a host. |
| $vCPU_{vm} : \mathcal{V} \times \mathcal{T} \rightarrow C$ | vCPUs used by a VM in a given moment. |
| $net_{vm} : \mathcal{V} \times \mathcal{T} \rightarrow 2^{\mathcal{N}}$ | Subset of network interfaces used by a VM in a given moment. |
| $Host : \mathcal{V} \times \mathcal{T} \rightarrow \mathcal{H}$ | Informs the mapping of a VM to a host in a given moment. |
| $VM : \mathcal{H} \times \mathcal{T} \rightarrow 2^{\mathcal{V}}$ | Informs the subset of VMs mapped to a host in a given moment. It is the inverse of function $Host$. |
| $L_{cpu} : \mathcal{V} \times \mathcal{T} \rightarrow L$ | Load generated by physical CPU usage of a VM in a given moment. |
| $\Psi_{vm} : L \times \mathcal{H} \rightarrow W$ | Power consumption related to a given load on the vCPUs on a host. |
| $bw_{net} : \mathcal{N} \times \mathcal{T} \rightarrow B$ | Bandwidth usage of a network interface in a given moment. |
| $bw_{vm} : \mathcal{V} \times \mathcal{T} \rightarrow B$ | Bandwidth usage of a VM in a given moment. |
| $L_{net} : \mathcal{V} \times \mathcal{T} \rightarrow L$ | Load generated by the networking operations of a VM in a host in a given moment. |
| $\Omega : B \times \mathcal{H} \rightarrow L$ | Load generated on a host related to bandwidth usage. |
| $\Psi_h : L \times \mathcal{H} \rightarrow W$ | Power consumption related to a given load on the dedicated hypervisor vCPU on a host. |
| $Cost : \mathcal{V} \times (\mathcal{T} \times \mathcal{T}) \rightarrow E$ | Total energy consumption of a VM in a given period of time. |
| $P^h_{total} : \mathcal{H} \times \mathcal{T} \rightarrow W$ | Total power consumption measured for a host in a given moment. |
| $P^h_{mngt} : \mathcal{H} \times \mathcal{T} \rightarrow W$ | Management consumption of a host in a given moment. |
| $P^{vm}_{mngt} : \mathcal{V} \times \mathcal{T} \rightarrow W$ | Management consumption related to a VM in a given moment. |
| $P^h_{min} : \mathcal{H} \rightarrow W$ | Minimum power consumption of a host. |
| $P^{vm}_{min} : \mathcal{V} \times \mathcal{T} \rightarrow W$ | Minimum power consumption of a host attributed to a VM in a given moment. |
| $P^{vm}_{cpu} : \mathcal{V} \times \mathcal{T} \rightarrow W$ | Power consumption generated by physical CPU usage of a VM in a given moment. |
| $P^{vm}_{net} : \mathcal{V} \times \mathcal{T} \rightarrow W$ | Power consumption generated by network operations of a VM in a given moment. |

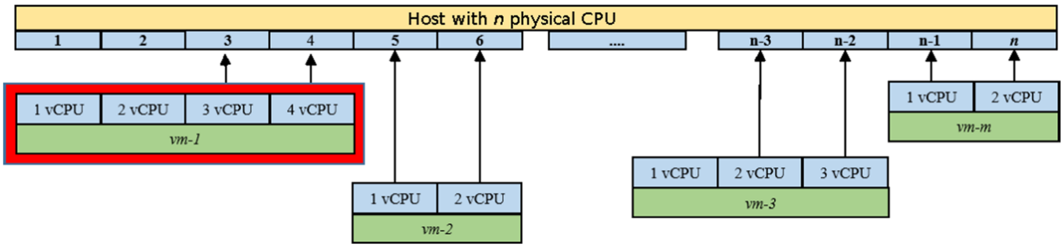



**Fig. 9** Example of vCPUs sharing a set of physical CPUs. The physical resource is multiplexed between vCPUs by the hypervisor scheduler. In this example, *vm-2* and *vm-m* have all vCPU assigned on the physical CPU, while *vm-1* and *vm-3* are partially assigned

Generalizing, lets consider an example in which the number of provisioned vCPUs is greater than the physical CPUs. In this scenario, the VMs's guest operating systems may have scheduled some processes to vCPUs, but the hypervisor did not map them in the physical processors. Clearly, this fact is common in resource sharing in IaaS clouds: a host can provision a number of vCPUs greater than the number of physical processors, and consequently, VMs can have vCPUs without real mapping, as mapping is dynamically performed by the hypervisor's scheduler.

Although other distribution criteria can be used (e.g., number of active VMs in the host, vCPUs allocated to VMs), PSVE aims at cost sharing proportional to CPU energy consumption, which represents the effective hosted application workload.

## 4.2 Proportional-Shared Energy Consumption

As discussed in Section 3.5, CPU consumption of a virtualized cloud server can be individualized per VM or collectively accounted and shared. Following the organization depicted in Fig. 3, the minimum CPU energy consumption and the management consumption are classified as collective, and must be proportionally shared among hosted VMs. Complementary, CPU and networking energy consumption resulting from VM workload represents the individual consumption.

The effective physical CPU usage from a VM $i$ at time instant $t$ is given by $L_{cpu}(i,t)$. In this sense, the energy cost (in Joules) of a VM $i$ for time period $T$ is given by (2).

$$Cost(i,T) = \int_{T_{start}}^{T_{end}} \left(P^{vm}_{cpu}(i,t) + P^{vm}_{net}(i,t) + P^{vm}_{min}(i,t) + P^{vm}_{mngt}(i,t)\right) dt \tag{2}$$

First of all, function $P^{vm}_{cpu}$ quantifies the power related to the processing workload of a VM. As depicted in (3), this is given by the load of the VM and the power consumption related to this load on a given host. Function $\Psi_{vm}$ (and $\Psi_h$) gives the calibration values obtained for a given host (as discussed in Section 3). It is worthwhile to observe that a load can have a different power consumption according to its host and source.

$$P^{vm}_{cpu}(i,t) = \Psi_{vm}(L_{cpu}(i,t), Host(i,t)) \tag{3}$$

The power consumption generated by network operations for a given VM is computed as given in (4). It is composed by the CPU server usage required for the network operations for this VM (5) and the power this CPU usage consumes on the host. The CPU server usage is given by a correlation of the network bandwidth used by the VM and the load it generates on the host, as illustrated in (5). Finally, the network bandwidth of a VM is represent in (6) as the sum of the bandwidth of its network interfaces in a given moment. It is worthwhile to mention that based on the experiments presented in Section 3.4, the consumption related with networking operations is influenced by the number of communicating VMs (Figs. 7 and 8), specially for communications performed atop a shared host.

$$P^{vm}_{net}(i,t) = \Psi_h(L_{net}(i,t), Host(i,t)) \tag{4}$$

$$L_{net}(i,t) = \Omega(bw_{vm}(i,t), Host(i,t)) \tag{5}$$

$$bw_{vm}(i,t) = \sum_{j \in net_{vm}(i,t)} bw_{net}(j,t) \tag{6}$$

As discussed in Section 3, a hosts minimum power consumption ($P^h_{min}$) is a constant value, obtained through a calibration method with energy measuring equipment in the target host. The fraction of minimum consumption attributed to a VM is quantified as indicated by (7). $P_{min}^{vm}$ represents the power consumption attributed to a VM, while $vCPU_h$ and $vCPU_{vm}$ represent the number of available virtual processors in the host and the number of vCPUs provisioned for the VM, respectively.

$$P_{min}^{vm}(i,t) = P_{min}^{h}(Host(i,t)) \times \frac{vCPU_{vm}(i,t)}{vCPU_h(Host(i,t))} \quad (7)$$

A virtualized server has a management power consumption required for keeping services and VMs running. However, this consumption is driven by the hypervisor and can vary for different frameworks. Thus, (8) quantifies the management consumption for a host $k$ by subtracting the consumption of all VMs hosted by it from its total power consumption. In this equation, $VM(k,t)$ indicates the subset of VMs that are hosted by $k$ at a moment $t$. Afterwards, for sharing the management consumption among hosted VMs, Equation (9) describes $P_{mngt}^{vm}$, which follows the goal of performing proportional sharing among hosted VMs.

$$P_{mngt}^{h}(k,t) = P_{total}^{h}(k,t) - \sum_{i \in VM(k,t)} \big(P_{min}^{vm}(i,t) + P_{net}^{vm}(i,t) + P_{cpu}^{vm}(i,t)\big) \quad (8)$$

$$P_{mngt}^{vm}(i,t) = P_{mngt}^{h}(Host(i,t),t) \times \frac{vCPU_{vm}(i,t)}{vCPU_h(Host(i,t))} \quad (9)$$

### 4.3 Integrating PSVE with Pricing Models

The energy cost per VM is constituted by the sum of the individualized costs. The result of using PSVE is given in energy consumption units (e.g., Joules). It is relevant to mention that PSVE only addresses the energy component of the total cost equation of a provider. The total resulting cost can include costs with support, depreciation, among others [1]. However, these costs are usually fixed and independent from VM processing load. Therefore, conversion to monetary cost is necessarily parameterized by the resource provider. Pricing is an option of the IaaS provider, which can attribute distinct values by time slots, regions or data centers. Although a deep study on pricing models is not the focus of the present work, we point-out a way for integrating PSVE with existing public cloud providers in order to highlight the differences between our proposal and current approaches.

Besides considering the energy consumption from CPU server, the final price for a given IaaS service (VM $i$, for instance) should consider elements from cooling, hardware depreciation, IT support and management operations, as represent by $\alpha_{mngt}$ in (10). In this equation, $Price_{vm}$ accounts for a Joules-to-money conversion, while $Price_{final}$ represents the final price delivered for a tenant related to reservation time period $T$.

$$Price_{final}(i,T) = \alpha_{mngt} + Price_{vm}(Cost(i,T)) \quad (10)$$

As we can see, the model can be incorporated by current public providers (e.g., Amazon EC2, Microsoft, Google, Rackspace, and Dualtec) that offer IaaS services. In this sense, Section 5 presents a case study with the Amazon EC2 provider, exemplifying PSVE's applicability.

## 5 Integrating PSVE with a IaaS Provider

In this section, we discuss how PSVE can be integrated with existing IaaS providers. It is worth to mention that PSVE deepens the energy-aware analysis of a DC consumption and can be potentially integrated with any pricing scheme. In this sense, we analyzed the price sheet of IaaS public providers, such as Google Compute Engine[6], Microsoft Azure[7], RackSpace[8], and Amazon EC2[9]. In brief, the price sheet is indexed by number of vCPUs, memory and storage types combined with extra management and provisioning services (*e.g.*, load balancing, firewall). Each requesting tenant can deploy a set of VMs with different pre-defined flavors.

Considering the pricing models, the IaaS providers provide IaaS based on *pay-as-you-go*, *pay-as-you-use*, or both. The first one is used to VM provisioning with monthly fixed prices, independently of VM workload

[6]https://cloud.google.com/compute/pricing

[7]https://azure.microsoft.com/en-us/pricing/

[8]https://www.rackspace.com/cloud/servers/pricing

[9]https://aws.amazon.com/ec2/pricing/

and status (active or not). The second model, *pay-as-you-use*, is applied for hourly provisioned VMs resulting on a non-fixed monthly price. In this case, the monthly price reflects the VM status (e.g., active, suspended), storage data, networking traffic, however, the final price is unaware of VM workload and energy consumption.

We can argue that both models perform an implicit accounting of physical server energy consumption. Notwithstanding, the model for accounting the resulting energy cost is not disseminated to tenants, and hardly considers a proportional share guided by VM workload. Following this line, on a *pay-as-you-use* scenario, two VMs provisioned with identical types are charged by a common and pre-defined price, even while having distinct workloads, networking profiles and consequently energy consumption.

It is a fact that the composing terms of IaaS providers price and cost models are sensitive information, not disseminated for requesting tenants. Thus, to perform an integration of PSVE cost model with IaaS providers' pricing scheme, we present a discussion on how to price VM instances based on VM processing energy consumption and VM networking energy consumption. In a nutshell, energy consumption is the key integration term.

### 5.1 Assumptions on Hardware Configuration

In our case study, we assume that our VM type is same of *m4.4xlarge*[10]. We chose it for having a similar hosting hardware configuration as the experimental calibration performed on Section 3. It is important to highlight that access to physical hardware on public cloud providers is not allowed, and consequently the calibration for identifying hypervisor CPU usage and consumption is rendered impossible by a tenant. Due to the impossibility of direct energy consumption measurement on the *m4.4xlarge* architecture or any another IaaS provider architecture, allied with the similarity of the physical hardware investigated on Section 3, we establish the premise that the minimum and core energy consumption are equivalent to those identified in Section 3. In this sense, 116 watts was assumed as the minimum energy consumption of *m4.4xlarge* instances.

### 5.2 Pricing a VM by Processing Energy Consumption

Among the available public IaaS providers, we decide to use public price sheet of Amazon EC2 for exemplifying the application of PSVE for extending a VM pricing model. For that, the current percentage of energy consumption considered by Amazon EC2 for composing the price sheet must be identified. As cost and price models of commercial providers are classified information, we apply some ranges based on public DCs for proposing approximated values. DC monitoring literature highlighted that 52% of a DC energy consumption is generated by IT equipment, while 15% of this value is attributed to CPU consumption [13, 14].

Following this line, we analyzed Amazon EC2's price sheet and considered three scenarios: 5%, 10% or 15% of a hourly reservation price is related to CPU energy consumption. In other words, this analysis compares the application of such constant values with a proportional distribution performed by PSVE. Referring to (10) (Section 4.3), 5%, 10% or 15% represent constant values for $Price_{vm}$ independently of the VM workload (as currently adopted by public providers).

On Amazon EC2's price sheet, there is a significant difference of values among Data Centers (DCs). This difference is justified by geographical location, management cost, but mostly by tax and kilowatt-hour price. The VM pricing discussed in this section is based on the performed prices for US West Data Center (DC) located in Oregon, US. For this DC, the kilowatt-hour price for commercial supplying is US$ 0,10[11].

The results of applying PSVE to a IaaS provider inspired in Amazon EC2 prices are depicted in Fig. 10. For that, the $x$ axis gives the VM CPU workload (0% to 100%) while the $y$ axis indicates the VM price resulted from CPU energy consumption (US$/hour). In order to represent the impact of energy consumption on the traditional price model, three constant parameters were used (5%, 10%, and 15%). The variable line represents the proportional sharing performed by PSVE. The vertical line indicates the typical CPU usage of data center servers, usually between 10% and 50%, with 30% on average [5, 31].

Considering the case in which traditional price model uses 15% for representing the energy consumption,

[10] http://www.ec2instances.info/?filter=m4.4xlarge.

[11] http://www.statista.com/statistics/263492/.

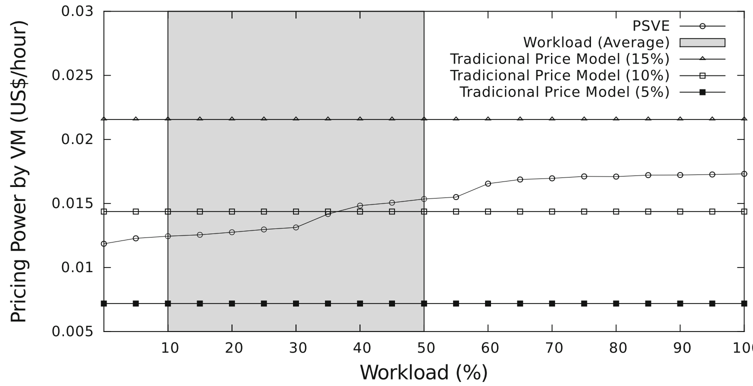



**Fig. 10** Integrating PSVE with a Public IaaS Provider pricing model. On the traditional price model, the pricing power by VM does not change due to the workload. On the other hand, PSVE adapts the VM price according to the workload

the resulting VM price is above the real energy consumption, while the opposite is observed for 5%. It is important to highlight in both cases that the traditional price model applies constant prices, independently of the VM energy consumption related to its management or processor / network usage.

A special case is observed in 10% scenario: the beneficiary depends on the VM effective workload. For workloads below 10%, the IaaS cloud provider is overcharging the tenants, whereas for workloads greater than 40%, the tenants are undercharged. It means that after 40%, independent of the VM workload, a tenant pays a constant value regardless of the effective energy usage.

Although such constant conditions may be justified by private business policies, they incentives a non-green usage of virtualized resources. Indeed, PSVE cost model enables an effective pricing based on CPU energy consumption. Following this line, besides the monetary incentive, tenants have an implicit motivation to control internal VM workload and CPU usage.

### 5.3 Pricing a VM by Networking Energy Consumption

We adopted Amazon EC2 price sheet to charge network traffic. This sheet is based on dollars per GB transferred. The energy profile identified on the calibration phase (Section 3) is used to account the energy consumption induced by network transfers between VMs. In short, three networking operations are charged: *(i)* data transfer in to VMs; *(ii)* data transfer out from VMs to Amazon services; *(iii)* data transfer out from VMs to the Internet. The prices for operations *(i)* and *(ii)* oscillate between US$ 0.00/GB and 0.02/GB, while for operation *(iii)* are accounted between US$ 0.05/GB and 0.09/GB. By observing the details of each operation[12], it can be concluded that Amazon EC2 has no differentiation by the hypervisor processing resulted from VM network operations.

The discussion presented on Section 3.4 indicated two aspects that impact on energy consumption resulted from networking operations: the aggregated bandwidth and the number of communicating VMs hosted by a server. Thus, a different energy consumption was observed for VM pairs hosted on a single node or spread atop a DC. This processing cost tends to increase if the packet inspection performed by the hypervisor is more rigorous [11, 20]. Therefore, the energy consumption arising from VMs communication must be accounted and proportionally shared among hosted tenants.

The first step for discussing the integration of PSVE with Amazon EC2 considering the network operations is to define the communication profile of a VM-hosted application. In this sense, for composing the results depicted by Figs. 11 and 12, the data transfer ranges from 30 to 308 TB/month for VMs. Both figures quantify the price of network traffic (US$) per month.

Figure 11 presents the results for VMs hosted by a single server. In this scenario, the traditional Amazon EC2 price sheet indicates a cost of US$ 0, 00/GB (operations are not charged). As discussed in Section 4, PSVE relies on a calibration to effectively account the CPU usage (and consequently the energy consumption) of a server for provisioning VMs

[12] Amazon EC2 networking prices: https://aws.amazon.com/ec2/pricing/.

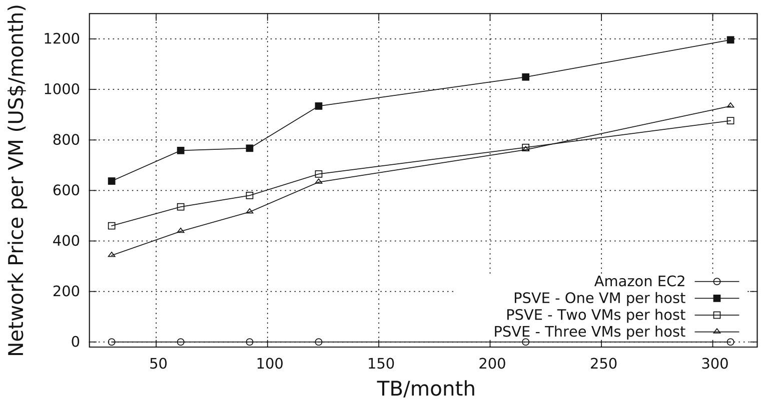



**Fig. 11** Pricing of Network Traffic in Amazon EC2 into Amazon EC2. Amazon EC2 has no fees to data transfer in to VMs. However, the data transfer between VMs affects the energy consumption of hypervisor. This energy consumption is considered in the VM price when using PSVE

connectivity. Moreover, the number of hosted and communicating VMs impacts on hypervisor processing. For representing different values retrieved by Ω (used in (5)), three scenarios of VMs connectivity were considered: one, two and three VMs per host. The total data monthly transferred for all scenarios is indicated in the horizontal axis, while the corresponding price is given by the vertical axis. A first look indicates that when only one VM is hosted by a server, all networking prices are directed to this tenant. Following this line, when the number of hosted VMs is increased, the aggregated bandwidth consumption remains constant (TB/month). However, the price is proportionally divided among all tenants based on hypervisor energy consumption.

The second scenario (Fig. 12) considered the expensive case on Amazon EC2, operation *(iii)*, data transfer out from VMs to Internet. In this scenario, Amazon EC2 prices vary between US$ 0, 05/GB and 0, 09/GB. Specifically, the provider applies 5 classes of prices for composing the final billing: *(a)* no charge up to 1GB/month; *(b)* US$ 0.09 by GB for next 10 TB/month; *(c)* US$ 0.085 by GB for next 40 TB/month; *(d)* US$ 0.07 by GB for next 100 TB/month; and *(e)* US$ 0.05 by GB for next 350TB/month.

Following the method applied to analyze operations *(i)* and *(ii)*, three configurations of VMs allocations were composed: one, two and three VMs per physical host. In this case, the TB/month usage for a single VM is up to 300TB.

It is worthwhile to note in Fig. 12 that PSVE can increase the final price of data transfers performed from VMs to Internet. Certainly, for all bandwidth configurations, Amazon EC2 is charging only the transferred packet, without considering the processing overhead introduced by VM networking operations. Finally, the CPU energy cost generated by networking operations is decreased when more VMs are hosted on a single node.

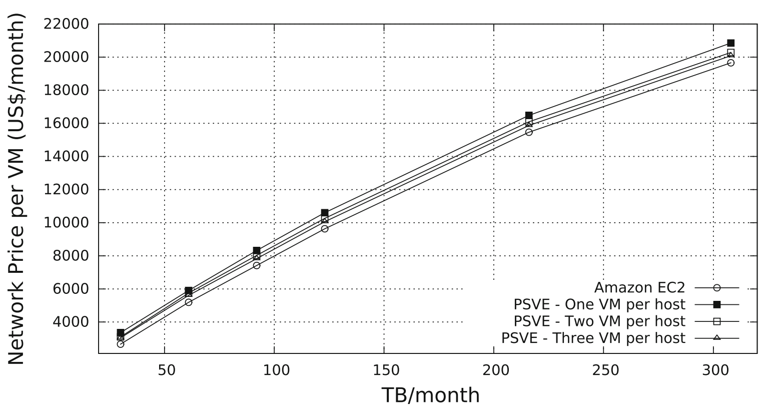



**Fig. 12** Pricing of Network Traffic in Amazon EC2 to Internet. All cases have similar prices, however PSVE can increase the final price of data transfers performed from VMs to Internet

### 5.4 Discussion and Key Observations

The case study presents beneficial situations for the IaaS cloud provider and tenants. The key goal of PSVE is to enable a proportional cost sharing based on VM energy consumption following the *pay-as-you-use* charging model. As a matter of fact, the application of PSVE is economically viable for costumers and providers independently if the VM-hosted application is CPU-bound or network-bound.

*A tenant should be charged by the effective CPU energy consumption of hosted VMs* The first analysis (Fig. 10) considers the CPU consumption of VMs. The equilibrium point between the traditional price model with and without PSVE is found at a configuration with 10% of the cost model dedicated to CPU energy consumption and 35% of VM vCPU workload. After this point, the application of PSVE is highlighted for composing the final cost. However, assuming that the cost model must account only for what is effectively used (*pay-as-you-use*), PSVE is the adequate choice.

*The energy consumption generated by networking operations must be addressed by cost models* The second analysis focusing on energy consumption of network operations (Figs. 11 and 12) indicated that an IaaS cloud provider must account for the hypervisor processing for composing the pricing model. It can be concluded that the cost for transferring large amounts of data, even internally on a DC, generate CPU energy consumption, and consequently, financial expenditure on the provider. Finally, following the *pay-as-you-use* principle, PSVE enables a proportional cost share of network operations among hosted VMs.

*Tenants can have an implicit incentive to process and communicate just when needed* Different from fixed provisioning in which a VM has a constant price during its lifetime, PSVE enables the cost accounting by its real CPU usage. In this context, a user has an implicit benefit for controlling the internal VM processing and data transfer. Techniques for decreasing the VM processing (e.g., reformulation of CPU-bound process) can be combined with networking optimizations (e.g., efficient communication libraries and strategies) to avoid high costs.

*PSVE is a first step towards energy-aware cost models* The proportional share of CPU energy consumption proposed by PSVE is a first step towards energy-aware VM provisioning. Although the case study highlighted positive aspects of PSVE, in future, it must be combined with traffic engineering approaches and proposals to address energy-aware routing and switching. On virtualized DCs, any networking entity can be virtualized to attend tenants requirements. In this sense, the quality-of-service provisioning increases the energy consumption for deploying virtualized routers and switches.

## 6 Considerations

Several important aspects were revealed in this article through the analysis and dissection of the elements related to the energy consumption of Virtual Machines (VMs) on IaaS clouds. The first aspect refers to the impression of the existence of just the cost of the running a VM when, in fact, there are shared costs that are sensitive to the simple fact of keeping a host running without any active VMs. Another relevant aspect stems from the energy consumption for network operations when both VM are on the same host, which, although providing higher transfer rates, generates a considerable amount of energy consumption. Finally, our studies and analysis show that processing and network energy consumption are relevant in pricing and may result in the redefinition of some model categories of IaaS providers to change from *pay-as-you-use* to *pay-for as-you-go*. In this sense, it is clear that the current charging model of Amazon EC2 fits more in a *pay-as-you-go* model than *pay-as-you-use* when taking into account the energy consumption.

Our case study (Section 5) presents how PSVE provides a fair cost model and is closer to the *pay-as-you-use* model than the model used by current public IaaS providers service. Although we apply PSVE for a price comparison with Amazon EC2, we want to point out that its purpose is not limited to pricing only. DC administrators can use the model to provide information as a basis for other approaches of accountability and even to evaluate the efficiency of their services/applications. Currently, there is an effort to develop applications optimized for servers in order to use less resources with an aim of becoming more scalable and consuming less energy. In this context, PSVE is a powerful tool to monitor and create tracking baselines.

A weak point of the solution is the need to run the calibration process to each pair of hardware and hypervisor, which can increase the model complexity if there is considerable hardware heterogeneity coupled with the continuous replacement of DC infrastructure. Changes on hypervisor software may also require running the calibration process again, even with the same hardware.

However, despite the aforementioned limitations, we note that PSVE provides a significant contribution towards cost sensitive models related to energy consumption. The model is flexible and robust, allowing adjustments of various aspects at the same time it enables to be evolved and improved.

**Acknowledgements** The authors would like to thank LabP2D (http://labp2d.joinville.udesc.br) for providing the testbed resources and technical support, and the Santa Catarina State University (UDESC) research funding program.

## References

1. Al-Roomi, M., Al-Ebrahim, S., Buqrais, S., Ahmad, I.: Cloud computing pricing models: a survey. Int. J. Grid Distributed Comput. **6**(5), 93–106 (2013)
2. Aldossary, M., Djemame, K.: Energy consumption-based pricing model for cloud computing, September 2016. This is an author produced version of a paper given at and published in 32nd UK Performance Engineering Workshop
3. Avelar, V., Azevedo, D., French, A.: Pue (tm): A Comprehensive Examination of the Metric. Technical report, The Green Grid (2012)
4. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. SIGOPS Oper. Syst. Rev. **37**(5), 164–177 (2003). https://doi.org/10.1145/1165389.945462
5. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. Computer (2007)
6. Begum, S., Khan, M.K.: Potential of cloud computing architecture. In: 2011 International Conference on Information and Communication Technologies (ICICT), pp. 1–5. IEEE (2011)
7. Beloglazov, A., Buyya, R.: Energy efficient resource management in virtualized cloud data centers. In: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 826–831. IEEE Computer Society (2010)
8. Chawla, C., Chana, I.: Strategy-proof pricing approach for cloud market. arXiv:1506.06648 [cs] (2015)
9. Cheng, L., Rao, J., Lau, F.C.M.: vScale: automatic and efficient processor scaling for SMP virtual machines. In: Proceedings of the Eleventh European Conference on Computer Systems, EuroSys '16, pp. 2:1–2:14, ACM, New York (2016)
10. Coroama, V., Hilty, L.M.: Energy consumed vs. energy saved by ICT - a closer look. In: 23rd Int. Conf. on Informatics for Environmental Protection (2009)
11. Cronkite-Ratcliff, B., Bergman, A., Vargaftik, S., Ravi, M., McKeown, N., Abraham, I., Keslassy, I.: Virtualized congestion control. In: Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference, SIGCOMM '16, pp. 230–243. ACM, New York (2016)
12. David, M.P., Schmidt, R.R.: Impact of ashrae environmental classes on data centers. In: Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2014 IEEE, 09 (2014)
13. Dayarathna, M., Wen, Y., Fan, R.: Data center energy consumption modeling: A Survey. IEEE Commun. Surv. Tutorials **18**(1), 732–794 (2016)
14. ENP: Energy logic: Reducing data center energy consumption by creating savings that cascade across systems (2008)
15. Feller, E., Rohr, C., Margery, D., Morin, C.: Energy management in Iaas clouds: a holistic approach. In: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 204–212. IEEE (2012)
16. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: enabling scalable virtual organizations. Int. J. High Perform. Comput. Appl. **15**(3), 200–222 (2001)
17. García García, A., Blanquer, I.: Cloud services representation using SLA composition. J. Grid Comput. **13**(1), 35–51 (2015)
18. Gmach, D., Rolia, J., Cherkasova, L.: Resource and virtualization costs up in the cloud: models and design choices. In: Proc. 41St IEEE/IFIP DSN (2011)
19. Han, Y.: Cloud computing: case studies and total cost of ownership. Information Technology & Libraries (2011)
20. He, K., Rozner, E., Agarwal, K., Gu, Y. (Jason)., Felter, W., Carter, J., Akella, A.: Ac/dc Tcp: virtual congestion control enforcement for datacenter networks. In: Proceedings of the 2016 Conference on ACM SIGCOMM 2016 conference, SIGCOMM '16, pp. 244–257. New York, ACM (2016). https://doi.org/10.1145/2934872.2934903
21. Iyengar, M., Schmidt, R., Caricari, J.: Reducing energy usage in data centers through control of room air conditioning units. In: Proceedings of the IEEE ITherm Conference in Las Vegas, pp. 1–11, 07 (2010)
22. Kansal, N.J., Chana, I.: Energy-aware virtual machine migration for cloud computing - a firefly optimization approach. J. Grid Comput. **14**(2), 327–345 (2016)
23. Kansal, S., Singh, G., Kumar, H., Kaushal, S.: Pricing models in cloud computing. In: Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies, ICTCS '14, pp. 33:1–33:5. ACM, New York (2014)
24. Kertesz, A., Dombi, J.D., Benyi, A.: A pliant-based virtual machine scheduling solution to improve the energy efficiency of iaas clouds. J. Grid Comput. **14**(1), 41–53 (2016)
25. Koomey, J.G.: Worldwide electricity used in data centers. Environ. Res. Lett. **3**(3), 034008 (2008)

26. Lee, Y.C., Zomaya, A.Y.: Energy efficient utilization of resources in cloud computing systems. J. Supercomput. **60**(2), 268–280 (2012)
27. Li, S., Lim, K., Faraboschi, P., Chang, J., Ranganathan, P., Jouppi, N.P.: System-level integrated server architectures for scale-out datacenters. In: Proc. 44th IEEE/ACM MICRO (2011)
28. Leong, B.G.L., Toombs, D.: Magic quadrant for cloud infrastructure as a service, worldwide. Gartner RAS Core Research (2015)
29. Mach, W., Schikuta, E.: Toward an economic and energy-aware cloud cost model. Concurrency Comput. Prac. Exp. **25**(18), 2471–2487 (2013)
30. Mastelic, T., Oleksiak, A., Claussen, H., Brandic, I., Pierson, J.-M., Vasilakos, A.V.: Cloud computing: Survey on energy efficiency. ACM Comput. Surv. **47**(2), 33:1–33:36 (2014)
31. Meisner, D., Gold, B.T., Wenisch, T.homas.F.: Powernap: eliminating server idle power. In: ACM Sigplan Notices, vol. 44. ACM (2009)
32. Mell, P., Grance, T.: The NIST definition of cloud computing. In: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology (2011)
33. Mobius, C., Dargie, W., Schill, A.: Power consumption estimation models for processors, virtual machines, and servers. In: IEEE Transactions on Parallel and Distributed Systems (2014)
34. Popek, G.J., Goldberg, R.P.: Formal requirements for virtualizable third generation architectures. Commun. ACM **17**(7), 412–421 (1974). https://doi.org/10.1145/361011.361073
35. Reza Rahimi, M., Ren, J., Liu, C.H., Vasilakos, A.V., Venkatasubramanian, N.: Mobile cloud computing: a survey, state of art and future directions. Mobile Netw. Appl. **19**(2), 133–143 (2014)
36. Rimal, B.P., Jukan, A., Katsaros, D., Goeleven, Y.: Architectural requirements for cloud computing systems: an enterprise cloud approach. J. Grid Comput. **9**(1), 3–26 (2011)
37. Ruck, D., Miers, C., Pillon, M., Koslovski, G.: Eavira: Energy-aware virtual infrastructure reallocation algorithm. In: 2017 VII Brazilian Symposium on Computing Systems Engineering, SBESC (2017)
38. Sharifi, L., Cerdà-Alabern, L., Freitag, F., Veiga, L.: Energy efficient cloud service provisioning keeping data center granularity in perspective. J. Grid Comput. **14**(2), 299–325 (2016)
39. Shuja, J., Bilal, K., Madani, S.A., Khan, S.U.: Data center energy efficient resource scheduling. Clust. Comput. **17**(4), 1265–1277 (2014). https://doi.org/10.1007/s10586-014-0365-0
40. Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: issues and challenges. J. Grid Comput. **14**(2), 217–264 (2016)
41. Srikantaiah, S., Kansal, A., Zhao, F.: Energy aware consolidation for cloud computing. In: Proc. of the Conf. on Power Aware Computing and Systems. USENIX (2008)
42. Srinivasan, M.K., Sarukesi, K., Rodrigues, P., Sai Manoj, M., Revathy, P.: State-Of-The-Art cloud computing security taxonomies: a classification of security challenges in the present cloud computing environment. In: Proc. ICACCI'12, ACM (2012)
43. Stallings, W.: Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud. 1st edn. Addison-Wesley Professional, Boston (2015)
44. Tang, S., Lee, B.-S., He, B., Liu, H.: Long-term resource fairness: towards economic fairness on Pay-As-You-Use computing systems. In: Proc. of the 28Th ACM Int. Conf. on Supercomputing (2014)
45. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost-Optimal scheduling in hybrid Iaas clouds for deadline constrained workloads. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 228–235. IEEE (2010)
46. Verma, A., Kaushal, S.: Cost-time efficient scheduling plan for executing workflows in the cloud. J. Grid Comput. **13**(4), 495–506 (2015)
47. Versick, D., Wassmann, I., Tavangarian, D.: Power consumption estimation of cpu and peripheral components in virtual machines. SIGAPP Appl. Comput. Rev. **13**(3), 17–25 (2013). https://doi.org/10.1145/2537728.2537730
48. Primet, P.V.-B., Anhalt, F., Koslovski, G.: Exploring the virtual infrastructure service concept in grid'5000. In: 20th ITC Specialist Seminar on Network Virtualization. Hoi An, Vietnam (2009)
49. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010)