



HAL
open science

Energy-proportional Profiling and Accounting in Heterogeneous Virtualized Environments

Mascha Kurpicz, Anne-Cécile Orgerie, Anita Sobe, Pascal Felber

► **To cite this version:**

Mascha Kurpicz, Anne-Cécile Orgerie, Anita Sobe, Pascal Felber. Energy-proportional Profiling and Accounting in Heterogeneous Virtualized Environments. Sustainable Computing: Informatics and Systems, In press, pp.1-36. hal-01633435v1

HAL Id: hal-01633435

<https://hal.science/hal-01633435v1>

Submitted on 13 Nov 2017 (v1), last revised 23 Feb 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Energy-proportional Profiling and Accounting in Heterogeneous Virtualized Environments

Mascha Kurpicz^a, Anne-Cécile Orgerie^{b,*}, Anita Sobe^a, Pascal Felber^a

^a*University of Neuchatel, Rue Emile-Argand 11 – 2000 Neuchatel, Switzerland
Email: {firstname.lastname}@unine.ch*

^b*Univ Rennes, Inria, CNRS, IRISA, 263 avenue du Général Leclerc – 35700 Rennes, France
Email: anne-cecile.orgerie@irisa.fr*

Abstract

The costs of current data centers are mostly driven by their energy consumption (specifically by the air conditioning, computing and networking infrastructure). Yet, current pricing models are usually static and rarely consider the facilities' energy consumption per user. The challenge is to provide a fair and predictable model to attribute the overall energy costs per virtual machine (VM) in heterogeneous environments. In this paper we introduce EPAVE, a model for *Energy-Proportional Accounting in VM-based Environments*. EPAVE allows transparent, reproducible and predictive cost calculation for users and for Cloud providers. **It provides a full-cost model that does not account only for the dynamic energy consumption of a given VM, but also includes the proportional static cost of using a Cloud infrastructure.** It comes with PowerIndex, a profiling and estimation model, which is able to profile the energy cost of a VM on a given server architecture and can then estimate its energy cost on a different one. We provide performance results of PowerIndex on real hardware, and we discuss the use cases and applicability of EPAVE.

Keywords: Energy-awareness; Energy Estimation; Accounting; Virtualized Systems; Heterogeneous Environments

*Corresponding Author: Anne-Cécile Orgerie

1. Introduction

The trend of computing in the Cloud grows, which consequently requires bigger data centers, more processing power and hence more CPUs. Whereas the hardware gets more and more energy-efficient the overall energy consumption of data centers increases. Actually, today's Cloud computing requires more electricity in the form of energy than entire countries such as India or Germany [1].

It is hence not surprising that energy represents one of the main cost factors of a data center. The major energy consumers are the air conditioning, the network infrastructure (routers, switches) and the servers [2]. However, these costs are rarely reflected in the attribution of energy consumption to a single consumer (e.g., a virtual machine).

As users share the same resources on a single node, most of the existing models concentrate on attributing the power consumption of this shared node to a single consumer. For instance, how much of the CPU power consumption can be related to a VM [3], [4], [5]?

Our vision is to consider energy accounting on the data center level to enable pricing models where every user will pay for the actual usage of resources. The first challenge is to provide a fair attribution model that is predictable to provide incentives for energy-efficient computing in the Cloud. The second challenge is to consider the *mobility* of VMs. A VM can be easily spawned on a different node, which might have different hardware specifications. These different hardware specifications might lead to different energy behavior.

In this paper we tackle these two challenges, by

1. showcasing EPAVE (*E*nergy-*p*roportional *P*rofilng and *A*ccounting in *V*irtualized *E*nvironments) for realizing accounting of real energy costs of the data center to each client considering the major consumers and the entire facility costs and
2. extend EPAVE with PowerIndex that allows us to predict the energy consumption of the same VM on different hardware.

More specifically, we target energy proportional accounting for each virtual machine (VM) in a heterogeneous environment. EPAVE provides a full-cost model that does not account only for the dynamic energy consumption of a given VM, but also includes the proportional static cost of using a Cloud infrastructure (comprising air conditioning, servers' static power consumption, etc.).

Context. Currently the relation between IT infrastructure and facility energy costs are modeled with the Power Utilization Efficiency (PUE) metric. This metric is used to help operators on decisions regarding new hardware infrastructure. For instance, Google measures the PUE per site each three months¹ for each of its data centers. While the PUE is a useful metric for reflecting the overall efficiency of a data center, its applicability for the day-to-day operation of the data center is limited because it does not grasp the variability of the actual power consumption of the data center.

In a data center, the instant power consumption can be divided into static and dynamic parts. The static parts are the base costs of running the data center when being idle; the dynamic costs depend on the current usage. In an ideal case, the overall power consumption would be proportional to the utilization of the hardware (power proportionality). However, having non-negligible static parts, power proportionality is not yet achievable [6], [7]. Nonetheless, we can get closer to power proportionality by accrediting the static power parts to each application, depending on the time and the resources used. Since time plays a major role, we will focus on energy instead of power consumption (an instant measure). Hence, we talk about energy proportionality rather than power proportionality.

Challenges. Dynamic power consumption mainly depends on the resources which are used: computing, storage, networking resources. In the case of virtual environments, the hardware resources may be shared among different users and different virtual machines, if they run on the same host. In this context, a power-

¹<http://www.google.com/about/data-centers/efficiency/internal/>

aware model needs to estimate the relative utilization per user to attribute the dynamic costs of the physical resources to a particular VM.

The static costs have to be considered at different levels:

- at the data center level: power delivery components, cooling systems, other miscellaneous components such as data center lighting. This part is captured by the PUE.
- at the resource level: idle power consumption of servers and routers.

The main challenge we tackle in this paper is to divide the static costs among the users in a fair and predictable way, considering the utilization of the resources per VM. We have shown in [8] that a simplistic model is not enough for distributing the costs among a number of VMs as the static costs attributed to each VM would be highly dependent on the utilization of the same server (i.e. number of VMs). To ensure fairness among the users and predictability, our energy proportional accounting model is independent from the Cloud provider’s VM management (not in control of the users): a given VM size executing a given application will get the same static cost from the EPAVE model even if executed at different dates on different servers.

As for dynamic costs, they can vary significantly from one server architecture to a different one. Performance and energy consumption heterogeneity among the servers is inherent to Cloud data centers. Typically, 3 to 5 server generations, with a few hardware configurations per generation, are hosted at the same time on a data center [9]; and this hardware heterogeneity leads to an important variability in terms of server performance [10]. In this paper, we extend EPAVE with PowerIndex, as we believe that a fair cost attribution should not only hold for a homogeneous setup, but our vision is also to be able to predict the costs of the VM when running on different hardware.

Contribution.

In this paper, we cover the accounting of dynamic and static costs to VMs in heterogeneous data centers and introduce the following two tools:

- EPAVE, a power-aware attribution model for VMs taking into account the overall consumption of the data center hosting them
- PowerIndex, a profiling and estimation model for accounting the costs (in terms of utilization and execution time) of a VM when running on different nodes

Organization. This paper is organized as follows. Section 2 presents the related work. The EPAVE model for energy attribution is detailed in Section 3. Energy mapping with PowerIndex is described and evaluated in Section 4. We discuss the properties of both models in Section 5 and provide an outlook on the usage of EPAVE and PowerIndex in Section 6. Finally, Section 7 concludes the paper.

2. Related work

Benefiting from economies of scale, Cloud infrastructures can effectively manage their resources and offer large storage and computing capacities while minimizing the costs for users. However, the rapid expansion of these infrastructures led to an alarming and uncontrolled increase of their power consumption. For instance, in 2010, the services offered by Google were based on 900,000 servers that consumed an average of 260 million Watts [11].

Moving from instrumenting to modeling the energy consumption is a tough but necessary task in order to improve the energy efficiency of distributed infrastructures. It is indeed essential to understand how the energy is consumed to be able to design energy-efficient policies.

2.1. Resource-based models

Most of the models found in literature split the consumption of an entire server into the consumption of each component of the server [3] or consider that consumption is proportional to the load [12]. Several studies are focused on modeling the energy consumption of particular components: CPU [13] influenced by the frequency, voltage and workload, network card [14] with costs per

packet and per byte, and disk [15] driven by the rotational speed and the read and write operations.

However, we have shown in [16] the limits of these approaches for modeling the energy consumption of entire servers under various workloads. Concerning the experimental approaches found in literature, they mainly consider just one type of machine, or even only one type of application [16]. So, it is necessary to design unified models closer to reality. Concerning the consumption of entire infrastructures, the authors of [17] show that computing resources represent the biggest part in Clouds consumption. An alternative approach [18] shows that, depending on the studied scenario, the energy costs of the network infrastructure that links the user to the computing resources can be bigger than the energy costs of the servers.

As shown in [16], simple models are not convincing in the general case and especially for multicore architectures – which tend to become widespread. It is therefore necessary to depend on benchmarks for the development and validation of reliable energy cost models for these heterogeneous resources. These benchmarks need to propose several kinds of workloads: computation-intensive, disk-intensive, etc.

2.2. VM models

Virtualization adds another layer of complexity and software power models are needed because it is not possible to attach a power meter to a virtual machine [19]. In general, VMs can be monitored as black-box systems for coarse-grained scheduling decisions, e.g., as done with *Joulemeter* [4]. If we want to be able to do fine-grained scheduling decisions—e.g., with heterogeneous hardware—we need to be able to consider finer-grained estimation at sub-system level and might even need to step inside the VM.

Bertran *et al.* [20] propose an approach that uses a sampling phase to gather data related to *performance-monitoring counters* (PMCs) and compute energy models from these samples. With the gathered energy models, it is possible to predict the power consumption of a process, and therefore apply it to estimate

the power consumption of the entire VM. This is similar to the work presented by *BitWatts* [5], which is further capable of estimating the power consumption of a process running within a VM and supports CPU-specific features such as hyperthreading and turbo frequencies.

Another example is given by Bohra *et al.* in [3], where the authors propose a tool named VMETER that estimates the consumption of all active VMs on a system. A linear model is used to compute the VMs' power consumption with the help of available statistics (processor utilization and I/O accesses) from each physical node. The total power consumption is subsequently computed by summing the VMs' consumption with the power consumed by the infrastructure.

2.3. Idle power consumption

The idle power concerns the power consumed by an infrastructure which is powered on but not running any task. Typically, for a server, it consists of the energy consumed while idle, but fully powered on. This consumption depends on the hardware of the server, but it can also depend on the operating system installed on it as it is responsible for the background tasks running continuously on the server (like monitoring tasks). This power is usually not taken into account by VM-based models described in the previous section. For an entire data center, the idle power consumption includes all the power which does not depend on the workload.

Often only the power consumption of IT equipment is considered although air conditioning can consume 33% of the global power needed by a data center [21]. This cost can be reduced by free cooling techniques exploiting outside air [22]. The power consumption of such cooling techniques is tightly correlated to the weather, and thus vary over time even if the workload does not vary. Therefore, their power consumption, which is considered to belong to the static idle part, can vary over time.

Most of the studies do not use the same definition for the energy costs of the computing infrastructure: for instance, the network used to link the computing resources is not taken into account most of the time. In the same way, as sur-

veyed in [2] some works take into account only the dynamic consumption of the machines and not their static consumption (corresponding to the consumption when machines are powered on but idle) which can yet represent more than half of the total power consumption. In our context, we will consider all the equipment operated by the Cloud provider: the data centers (including cooling infrastructures) and the network links inside and between the data centers.

In addition to these considerations, simply measuring the power consumption of computing resources may pose problems of security and confidentiality as identified in [23]. Indeed, it is shown that by simply having access to the energy consumption of a cloud server, one can guess with high probability what type of application, among various possible, was running in its virtual machines. It is therefore necessary to consider an instrumentation of these platforms that guarantees the privacy of the user, of the provider regarding its machines, and of the applications. This is why our model do not rely only on direct measurements on the physical machines: because it would reveal too much information. For instance, it could reveal the server’s position in the rack as it influences the power consumption due to the air conditioning situation [16].

2.4. *Heterogeneous data centers*

Existing research often assumes that data centers consist of homogeneous hardware. However, this is not realistic in real-world data centers, as shown in [24]. We often find different generations of the same hardware in a data center, or even different types of hardware combined. This tendency led to the concept of *warehouse scale computers (WSCs)* [25] composed of diverse microarchitectures and configurations.

Running workloads on heterogeneous hardware can impact the energy efficiency of the entire setup. In previous work [26] we presented our study on the impact of energy patterns on scheduling decisions in heterogeneous data centers. We considered a very simple scenario where we placed ten workloads on 3 heterogeneous machines. Half of the workloads were writes of 30 seconds to the disk and the other half were CPU/memory-intensive factorial computations

of a large number. We compared the total energy consumption of the different scheduling scenarios. Only by placing the workload on a non-appropriate server increased the energy consumption at least by 12 % and up to 14 times more in the worst case, in comparison to the optimal placement, as shown in Figure 1. In order to place a workload correctly on the best-fitting available resource (i.e., physical machine), we need to know the power consumption of the machines for specific types of workloads.

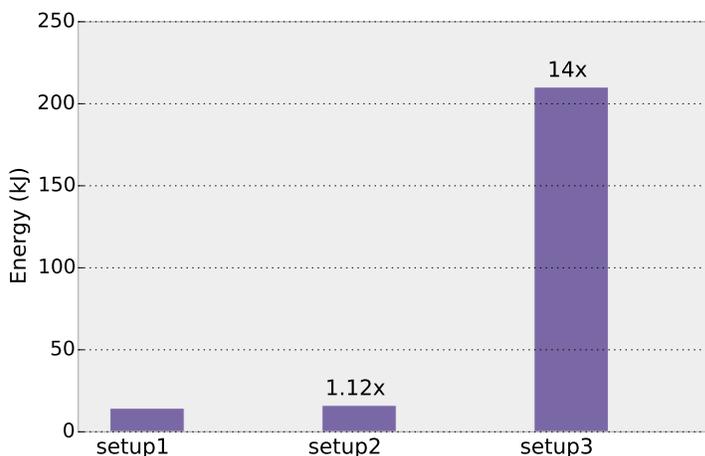


Figure 1: Different placement configurations in heterogeneous data centers have a large impact on energy consumption [26].

Paragon is a scheduler aware of heterogeneity and interference [9] [27]. It decides for a specific workload on which server configuration it will perform best (heterogeneity). It also considers how much interference it will cause to other applications and how much interference it can tolerate regarding its own shared resources. This approach mainly focusses on performance and preserving the QoS rather than power or energy efficiency.

Another existing approach focusses more on the power aspect: It defines an intelligent workload allocation method that efficiently maps workloads to the best matching platform [28]. The authors show that their approach signifi-

cantly reduces the overall power consumption of the data center (12-22% power saving over random allocation). In their evaluation, they measure power and performance across each platform and then extrapolate the measured data using a data center allocation simulator.

3. Energy attribution with EPAVE

The key idea of EPAVE is to attribute the data center’s static and dynamic costs (C) to each VM, which can be then used as a basis for several use cases as described later. As costs we consider the total consumption (C_{total}) during the execution of a VM in the context of a data center.

$$C_{total} = C_{static} + C_{dynamic}$$

The static costs comprise the idle consumption of each node and the idle consumption of the routers as well as induced consumption of the entire data center (routers, air conditioning, power distribution units, etc.). To cover the entire data center the Power Usage Effectiveness (PUE) has become the industry-preferred metric for measuring infrastructure energy efficiency for data centers [29, 30]. It is defined as the ratio of total facilities energy to IT equipment energy:

$$PUE = \frac{Total\ Facility\ Energy}{IT\ Equipment\ Energy}$$

Therefore, we will use the PUE to account for the consumption part exterior to the IT equipment itself which is already taken into account. As outlined in the data center industry survey conducted by Uptime Institute Network (a user group of large data center owners and operators) [30], the adoption of PUE is rising worldwide, and its measurement and improvement are widely targeted by the 1,000 surveyed data center operators and IT practitioners. That is why we believe that the PUE metric is easily accessible for Cloud providers for their data centers. From the PUE definition, for 1 Watt consumed by the IT equipment, the entire data center infrastructure consumes in fact $1 \times PUE$ Watts. Therefore,

the static costs of a data center have to be multiplied by the data center’s PUE:

$$C_{static} = \left(\sum^{\#nodes} C_{idle_{node}} + \sum^{\#routers} C_{idle_{router}} \right) \cdot PUE$$

The dynamic costs include the dynamic energy consumption part of the servers, routers and storage devices, which we can formulate as a weighted sum of the individual costs. The weights represent the resource usage of the current workloads, which can be between 0 and 1, where 1 means maximum utilization of the given resource and 0 means no utilization.

$$C_{dynamic} = \alpha \cdot C_{comp} + \beta \cdot C_{IO} + \gamma \cdot C_{net}$$

To attribute the overall costs to a single VM, we first need to distribute the idle costs in a fair and transparent manner. In many cases the idle costs (or idle power consumption) are only divided by the number of VMs [3]. However, for energy-proportional accounting it is necessary to consider the size of a VM, and in particular, its number of vCPU as CPU is the most consuming device in a server [2]. Inspired by the VM types chosen by Amazon we will differentiate VMs by the number of their assigned virtual CPUs (often proportional to their assigned memory volume). In addition we want to take into account heterogeneous data centers.

The dynamic costs are determined when the VM finishes by using the real resource utilization of the physical resources. The total costs are limited to the static costs as the lower bound, and on the maximum consumption as the upper bound. Reporting these bounds to the user makes the final VM’s costs predictable (bounded) and keeps the spirit of the pay-as-you-go manner although the dynamic part of the costs is in most cases smaller than the static parts (reflecting the reality of the power consumption of typical data center servers).

To sum up, the maximum costs of a VM grow with its size as shown by an illustrative example in Figure 2. In this experiment we were inspired by the Amazon VM sizes. The static costs, $C_{static}(VM)$, depend on the number of cores reserved by the VM, and the dynamic costs $C_{dynamic}(VM)$ on the actual

usage (in our case performed by the stress command). In more details, we model the static and dynamic costs per VM as described in the following paragraphs.

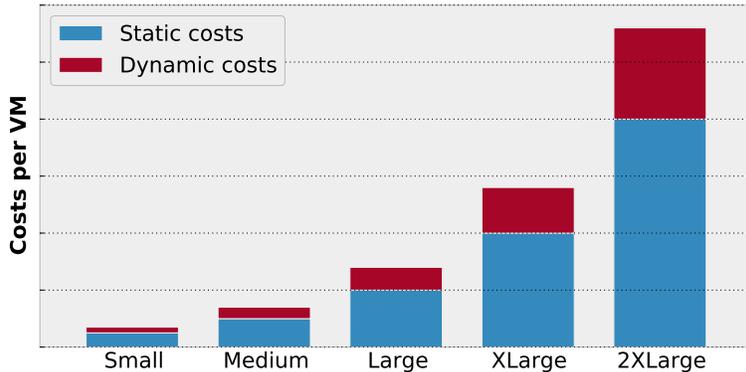


Figure 2: Example of maximum costs distribution among different types of VM

3.1. Static costs

For the static costs, we could define them depending of the VM size along all its resources: CPU, memory and disk. Yet, it would require a complex formula to weight the dimensions in order to reflect their impact on power consumption, and this weighting would depend on the hardware capacities of a given server. Such a formula would be difficult to instanciate in practice [31]. This is why we adopted a simple model similar to the one currently in application at Amazon [32]: the costs are proportionnal to the number of virtual CPUs assigned to each VM. As we want these costs to be static and independent from the hypervisor, we use a weighted averaged value of the idle power consumption of all the servers. This model is similar to the one currently in application at Amazon [32]: the costs are proportional to the number of virtual CPUs (vCPU) assigned to the VM:

$$C_{static}(VM) = \frac{\#vCPU(VM)}{\sum^{\#nodes} \#CPU(node)} \cdot C_{static}$$

3.2. Dynamic costs

The dynamic costs are hardware and application dependent and require monitoring. According to [33] the acceptance of dynamic models is increased if the costs are limited by an upper bound. Indeed, a VM cannot exceed the physical resources allocated to it (CPU, RAM and disk mainly), so the upper bound can be determined for each type of VM over each type of physical node. The actual dynamic costs per VM will be in the range of 0 (idle) and the maximum resource usage. The challenge is to attribute the maximum dynamic costs to a VM. In general, the dynamic costs of a VM are the measured or estimated energy consumption (E), which is the integral of the power consumption (P) measured/estimated per time unit (T).

$$C_{dynamic}(VM) = E(VM) = \int_0^T P(VM) dt$$

In general, a VM cannot consume more than the maximum dynamic costs of the entire server ($C_{dynamic}$). If the VM is co-located with other VMs it is necessary to split up the dynamic costs. Here, we use a very simple model to define an approximate upper bound for the costs of a VM, by using the number of cores the VM got assigned as a basis. Note that the focus on the number of vCPUs (ignoring the disk and network) is chosen because the number of vCPUs usually differentiates VM sizes offered by Cloud providers. Additionally, the CPU is one of the highest power consumers on a node and CPU-intensive applications are the most consuming ones [16].

$$0 \leq C_{dynamic}(VM) \leq \frac{\#vCPU(VM)}{\#CPU(node)} \cdot C_{dynamic}(node)$$

An alternative would be to use a software power estimation model that is capable of attributing the dynamic costs to a VM, such as VMeter [3], or Bitwatts [5].

3.3. Use Cases

In this section, we showcase how to calculate $C_{static}(VM)$ and $C_{dynamic}(VM)$ based on real-world experiments. Based on the real data we can use EPAVE

to estimate the costs of different use cases. For the experiments, we rely on selected nodes from the Grid’5000 cluster to which powermeters are attached [34]. Specifically, we performed the experiments on two kinds of nodes, *Taurus* and *Sagittaire*, whose characteristics are specified in Table 1. We further consider different sizes of VMs, which are inspired by the Amazon instances and shown in Table 2.

Table 1: Hardware characteristics of the selected systems

Hardware	Taurus: Dell PowerEdge R720	Sagittaire: Sun Fire V20z
Model	Intel Xeon e5-2630 (2.3GHz)	AMD Opteron 250 (2.4GHz)
Cores/Threads	2×6/12	2/2
RAM (GB)	32	2
TDP (W)	2×95	215
# Servers	16	79

3.3.1. Homogeneous setup

Figure 3 presents the costs $C_{total}(VM)$ for a homogeneous cluster with *Taurus* servers with 12 cores each. Their average idle power consumption is 95W per server. In a real setup the calculations need to include network costs and PUE, hence we need to add the costs for a number of switches (approx. 350W each) and multiply by the PUE (e.g., 1.22). In this specific example, we demonstrate the cost models based on the idle power of the servers as a matter of simplification for the calculations. The static costs per core are easy to compute: $95/12 = 7.92$. The dynamic part represents the maximal achievable dynamic power consumption when running the *stress* command. Together these costs represent the upper bound of costs per VM. We can see that the static costs increase proportional to the number of cores assigned to the VM and two VMs having together 12 cores will reach the same static costs as the machine itself. Hence, in a homogeneous setup EPAVE would fall back to a trivial model, where only the upper bound of costs $C_{total}(VM)$ have to be reported.

Table 2: VM types

Type	Medium	Large	XLarge	2XLarge
Number of cores	1	2	4	8

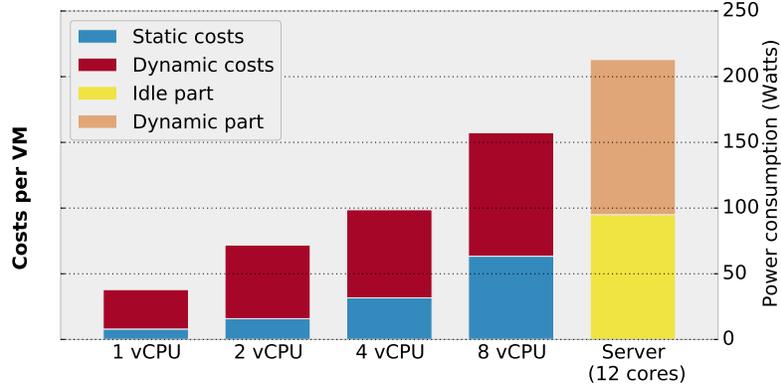


Figure 3: Example of maximum cost distribution among different types of VM for a homogeneous cluster

3.3.2. Heterogeneous setup

If we switch to a heterogeneous use case (as shown in Figure 4) and run again the *stress* command, the static costs are not proportional anymore to the number of virtual cores as the idle power of the machines might be unbalanced. We showcase the unbalanced scenario with experiments performed on two different kinds of servers, whose characteristics are summarized in Table 1. The two clusters are heterogeneous in terms of server architecture, but also in terms of number of nodes, and number of cores per node. The idle power consumption represents the average power consumption of a server over the entire cluster.

In this use case we can calculate the static costs for a one-vCPU VM as follows:

$$\begin{aligned}
 C_{static}(VM) &= \frac{1}{\sum \#nodes \#CPU} \cdot \sum^{\#nodes} C_{idle_{node}} \\
 &= \frac{16 \cdot 95 + 79 \cdot 215}{16 \cdot 12 + 79 \cdot 2} = 52.87
 \end{aligned}$$

These costs are more than 6 times higher than in the homogeneous case

with only *Taurus* nodes, but they represent half the costs of a cluster with only *Sagittaire* nodes. Therefore, heterogeneity among nodes leads to average static costs per virtual CPU which can be far from the costs per cluster. However, this is a healthy property of EPAVE: in order to cover the real energy costs with this accounting, the Cloud provider has to favor the utilization of the most energy-efficient servers. To provide incentives for clients to use the most energy-efficient setup, we introduce PowerIndex later.

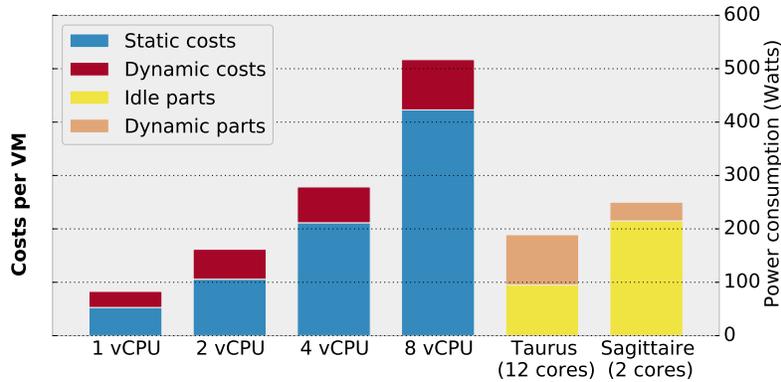


Figure 4: Example of maximum cost distribution among different types of VM for a heterogeneous cluster with unbalanced idle power for the server architectures

3.3.3. Underutilization of reserved resources

To show the applicability of our models, we performed experiments using real-world applications on a *Taurus* node. We installed Hadoop Yarn [35] on each of the nodes and ran *sort* and *wordcount* from the HiBench [36] benchmark suite. We run the workloads within a VM to be able to limit the number of cores they use in total. We started the VM once with only a single core, and once with all cores available. This experiment is the basis for three use cases, where we want to showcase the effects of underutilization of reservations. Note that the dynamic costs are always measured in real experiments and the static costs are predetermined based on the idle power consumption.

The workloads have different power consumption patterns as shown by the

example of the *wordcount* workload executed on all available cores in Figure 5. The idle power of the *Taurus* nodes is 95W. We also know the maximum total power of 220W, and 125W as basis for $C_{dynamic}$ for all reserved cores without idle power. These values can be predetermined and have to be collected only once per architecture. The actual dynamic costs of the workload vary between 0 and 100W over a runtime of 200 seconds. This shows the necessity of considering energy rather than power consumption, as we need to provide models that reflect the actual usage of the VM over time.

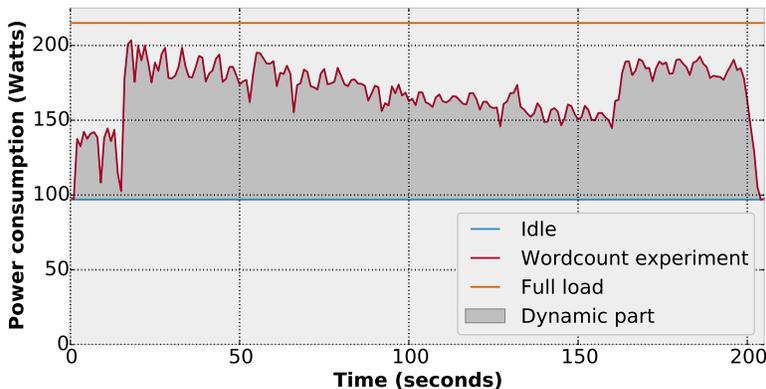


Figure 5: Power profile of the wordcount workload using all available cores

If we consider the pay-as-you-go model as a basis, a VM would cost according to its size (i.e., resources reserved) and according to the time used. The same idea is followed by EPAVE, but we consider both static and dynamic energy as a basis of costs. As an example, for the single core experiment, we calculate $C_{total}(VM)$ according to our model and fill it with values from our experiments.

$$C_{total}(VM) = C_{idle} * ratio_{vCores} * runtime + C_{dynamic}$$

As shown in Figure 6 the static costs for using only a single core are smaller. However, because the single core is used for a longer time span, the dynamic costs are much higher leading to higher total costs than if all cores are used and reserved.

Let us consider a use case where the workload is not optimized for paral-

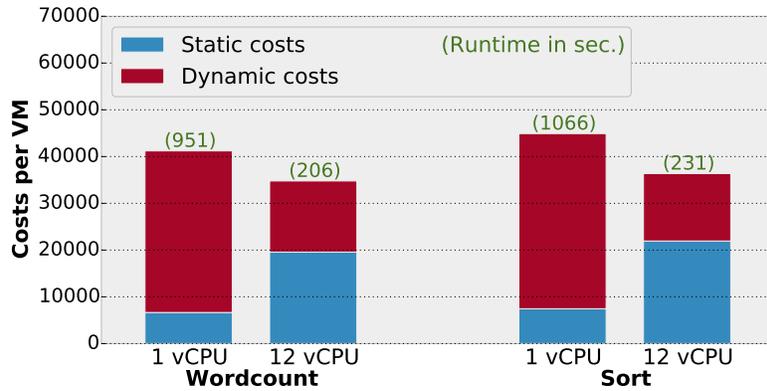


Figure 6: Costs of two parallel workloads with a reservation of one core and twelve cores

lelization, but still the reservation covers all of the cores. If a workload only uses a single core out of 12, the dynamic costs will not change in comparison to the former use case, however, the static costs are distributed among the number of cores served. Taking the dynamic costs of the former experiment as a basis, this would mean a significant increase in costs for the VM (see Figure 7). In an ideal case a user is encouraged to reserve resources according to the resources required and parallelization capabilities of the workload.

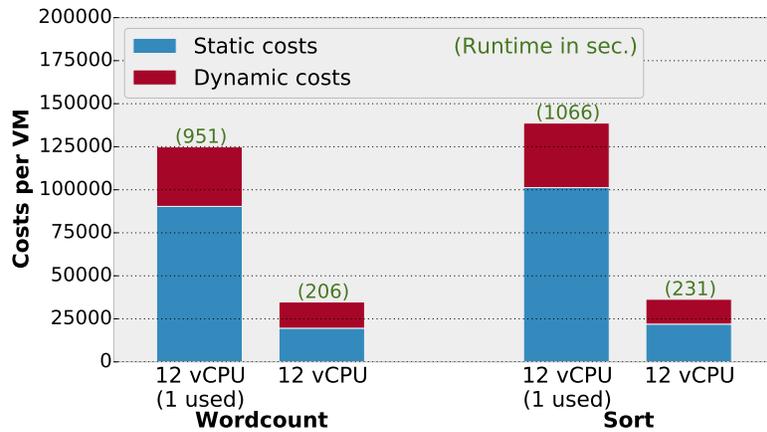


Figure 7: Costs of two workloads with underutilization of reserved cores

The runtimes of the former experiments are rather low and we assumed that

the reservation for a VM ends with the end of the workload. However, in reality most VMs are reserved for a given time span. For instance, if we consider the default minimum reservation of VMs of around 20 minutes the cost distribution for the same workloads changes and the results are depicted in Figure 8. Hence, if we reserve all cores for 20 minutes but only use them for the first few minutes the static costs exceed the dynamic costs and the single core reservation is much more advantageous.

With EPAVE it is possible for a user to identify such discrepancies and decide for what kind of reservation is useful. Another possibility is to use tools such as PowerIndex as introduced in Section 4 to provide insights on what costs should be expected when running in a heterogeneous setup.

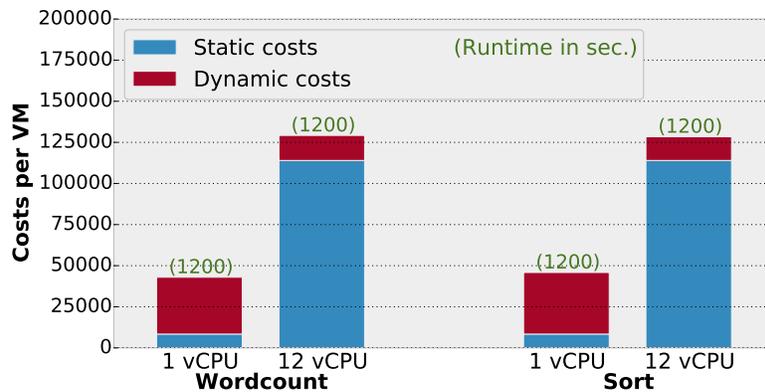


Figure 8: Costs of two workloads with predetermined reservation time of 20 minutes per VM

4. Energy mapping with PowerIndex

With EPAVE we are able to predict the costs of a VM in a coarse grained manner on a machine it should run on. For the final decision on where the user should run the VM, we introduce PowerIndex. PowerIndex is a tool to map different workload scenarios in a heterogeneous environment. Given a specific workload, PowerIndex provides an estimation of the potential energy consumption for execution on different types of machines. Usually, this would

require to profile a given application on all available types of machines. This is obviously not efficient and not practical. The goal of PowerIndex is to be predictable and lightweight, only requiring minimal profiling effort. PowerIndex is a tool that is based on offline and online profiling, especially applicable for repetitive applications. The offline profiling is performed only once per machine type and allows us to build a reference power profile and mapping between a reference machine and all other machines. In the online phase a new workload is only profiled on the reference machine and the power and utilization mappings are used for predicting the VM's approximate costs on all other machines to perform proper scheduling decisions. These characteristics of PowerIndex makes it a useful tool for Cloud users in order to rightly dimension their VMs for a given workload.

4.1. Offline Profiling

Our offline profiling approach consists of two main components: the Power Table and the Utilization Mapping.

4.1.1. Power table

The *Power Table* stores information about the offline power consumption for each machine type (Machines $M = m_1, m_2, \dots, m_m$). To set up the Power Table, we execute different microbenchmarks consisting of CPU and disk-intense workloads. The CPU-intense microbenchmark comprises *stress* and *cpulimit*. We stress the CPU with different intervals of CPU load from 5 to 100%, in steps of 5 %. For the execution of the workloads we want to cover both the user-space and the kernel-space utilization of the CPU and therefore cover %usr and %sys metrics. As a result we have a table of power consumption for %usr utilization intervals and for %sys intervals. The Power Table can be described formally for the profiling of a machine m_1 :

$$\text{Utilization } U = \{U_{usr}, U_{sys}\}$$

$$\text{Interval } I = \{(x,y),..\}; \forall x,y \in \mathbb{N}; x,y \in [0, 100]$$

$$\text{Power } P = \text{Watt} \subseteq \mathbb{N}$$

Under the condition $\forall U_i \in U : U_i \subseteq I$

Define Power Table PT_{m_1} (for sys and usr) for a machine m_1 :

$$\forall U_i \in U; \forall u_i \in U_i; f : U_i \times M \rightarrow P \quad f(u_i, m_1) = p_{i,m_1}$$

Given a filled power table, function f maps the power consumption to a utilization interval, e.g., $f(u_i, m_1) = p_{i,m_1} = f((0, 20), m_1) = 10W$.

4.1.2. Utilization Mapping Table

When a workload runs on one machine, it will most likely not have the same utilization on another machine. Therefore, we need a basic understanding of how utilization u_{m_1} translates to u_{m_2} . Out of simplicity we provide the mapping between a reference machine m_1 and all other machines and define the *Utilization Mapping Table*.

We define a function ut_{sys} to map the system utilization from m_1 to the other machines. Given the utilization on the reference machine for a specific workload, we want to know what utilization to expect on the other machines.

Utilization $U \subseteq \mathbb{N}$, $\forall u_i \in U, u_i \in [0, 100]$

$\forall u_i \in U, \forall m_i \in M$

$ut_{sys} : U \rightarrow U$

$ut_{sys}(u_{i,m_1}, m_i) = u_{i,m_i}$

The corresponding function for *usr* can be defined analogously.

4.2. Online Monitoring

The major part of our estimation is done in the offline profiling, which has to be only done once per machine type. The online monitoring is performed when a new workload arrives (that has not been logged already). PowerIndex requires that the new workload is run once for a configurable time on the reference machine and then uses the data from the offline tables to map the energy consumption on all other machines. The profiling time should be adapted to the expected workload. If the workload consists of repetitive phases, the profiling time should cover one or two phases.

To avoid duplicate profiling of workloads, we identify meta tags for each workflow and store it in a backend database. We chose this approach because we want to limit the profiling time of a workflow to as little as possible. This way, the system can learn any new workload and does not require any profiling in case of known workload.

4.3. Toy Example

To show the interaction between the offline and online profiling we consider a simple example. In a real data center, the workloads could be virtual machines or containers instead of simple workloads and the intervals would be chosen in a more fine-grained manner. Whenever a new unknown workload (w_1) arrives, we need to profile it on the reference machine and categorize it to obtain power values for each machine. If the workload w_1 is already known, we can just look up the required values in the database. PowerIndex holds the Utilization and Power Tables and we use them to get the final power consumption. Note that for this example we use a simplified set of Power and Utilization tables than described above due to space reasons. The utilization mapping assumes that machine m_2 has twice as many cores as machine m_1 .

Table 3: Power Table for %usr and %sys utilization for machine m_1 and machine m_2

%usr	Power m_1 (W)	Power m_2 (W)	%sys	Power m_1 (W)	Power m_2 (W)
0	0	0	0	0	0
25	7.5	10	2	1	2
50	15	20	4	2	4
75	22.5	30	6	3	6
100	30	40	8	4	8
			10	5	10

With the given Power Tables and Utilization Mapping Tables we can start to estimate a simple workload, as described below.

Workloads W

$$\forall w_i \in W : \exists! u_{sys} \in U_{sys} : p_{sys} = f(u_{sys}, m_1)$$

$$\forall w_i \in W : \exists! u_{usr} \in U_{usr} : p_{usr} = f(u_{usr}, m_1)$$

Table 4: Utilization Mapping Table from reference machine m_1 to machine m_2 for %usr and %sys

%sys m_1	%sys m_2	%usr m_1	%usr m_2
0	0	0	0
2	1	25	12.5
4	2	50	25
6	3	75	37.5
8	4	100	50
10	5		

$$p_{wi,mi} \in P$$

$$p_{idle} \in P$$

$$p_{wi,m1} = p_{idle} + p_{sys} + p_{usr}$$

In this toy example, we assume a very short workload and that we monitor it entirely. At each point in execution, the power required for this workload on machine m_1 is the sum of the power required for the *sys* utilization, for the *usr* utilization and the idle power. We assume that the reference machine has an idle power of 29W. The online profiling of workload w_1 on the reference machine m_1 results in the CPU utilization values as shown in Table 5.

Table 5: Utilization trace for the simple workload w_1 on machine m_1

Time (s)	%usr	%sys
1	50	6
2	25	6
3	25	6
4	100	0
5	100	0

If we lookup the power for the utilization measured in Table 5 in the Power Tables of the machine m_1 , we can easily compute the energy by summing the power values per second:

$$E_{m1} = (29+15+3)+(29+7.5+3)+(29+7.5+3)+(29+30+0)+(29+30+0) = 244J$$

The execution of workload w_1 on machine m_1 would thus cost, in terms of

energy, 244 Joules. To estimate the power on other machines, we need to map the measured utilization on m_1 to the utilization of the other machines using the Utilization Mapping Tables. Once we obtained the mapped utilization for machine m_i , we can get the power value from m_i 's Power Table.

$$\forall w_i \in W : \exists! u_{sys} \in U_{sys}: \text{measured util. on } m_1$$

$$u_{sys,mi} = ut(u_{sys}, m_i): \text{mapped util. on } m_i$$

$$p_{sys} = f(u_{sys,mi})$$

$$\forall w_i \in W : \exists! u_{usr} \in U_{usr}: \text{measured util. on } m_1$$

$$u_{usr,mi} = ut(u_{usr}, m_i): \text{mapped util. on } m_i$$

$$p_{usr} = f(u_{usr,mi})$$

$$p_{wi,mi} = p_{idle} + p_{sys} + p_{usr}$$

This constructs the utilization trace as shown in Table 6.

Table 6: Expected utilization trace for the simple workload w_1 on machine m_2

Time (s)	%usr	%sys
1	25	3
2	12.5	3
3	12.5	3
4	50	0
5	50	0

We can then lookup the power consumption for the mapped utilization values in the Power Tables for machine m_2 (Table 3). If we assume an idle power of 50W for machine m_2 , we get the following energy consumption:

$$E_{m_2} = (50+10+3)+(50+5+3)+(50+5+3)+(50+20+0)+(50+20+0) = 319J$$

We expect workload w_1 to cost 319 Joule on machine m_2 . This toy example assumes that the execution time for workload w_1 is the same on machine m_1 and machine m_2 . Otherwise, the execution time would be mapped similar to the utilization mapping presented in this toy example.

4.3.1. Real-World Use Case

In this subsection we consider a well known benchmark PARSEC on real hardware. We use again the *Taurus* and *Sagittaire* machines as defined in Table 1. These two types of machines have considerable differences in hardware (number of cores, CPU type, etc.) and therefore are very good examples for our experiments. We assume that the offline profiling, which is done once per architecture, has already been executed. Thus, the Power Tables and the Utilization Mapping Table are already available. We execute each benchmark for 20 s on the reference machine *Sagittaire*, and then look up the results from the offline profiling. We compare against the measured energy consumption with the power meters attached to the machines.

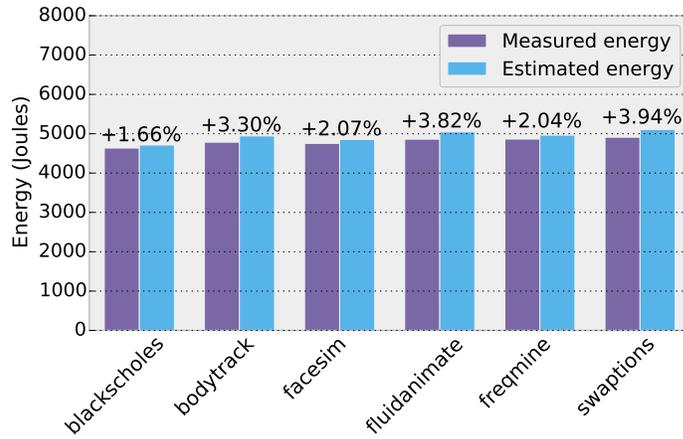


Figure 9: Energy estimation of the parsec workloads on the reference machine *Sagittaire*.

Figure 9 shows the energy estimation for the Parsec benchmarks on the reference machine. We observe errors of less than 4%.

Figure 10 depicts the energy estimation on the *Taurus* machine. In this case we need to consult the Utilization Mapping Table and the Power Table. The energy estimation for the *Taurus* machine is based only on the values measured on the *Sagittaire* machine. We encounter an underestimation with a relative

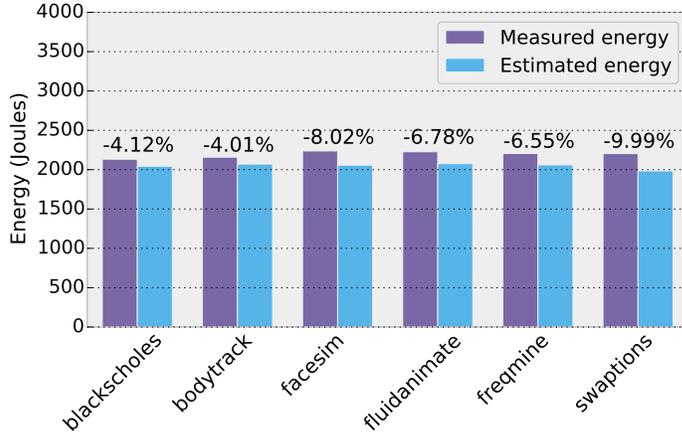


Figure 10: Energy estimation of the parsec workloads on the *Taurus* machine.

error that ranges between -4% in the best case and -10% in the worst case. The errors we encounter are acceptable in our use case. Errors up to 10% are common in the related work of power and energy estimation [5]. In our case we can even accept errors that are slightly higher since we do not rely on high accuracy.

5. Discussion

EPAVE keeps the philosophy of the Cloud: the pay-as-you-go model but based on energy consumption. The costs of a VM indeed depend on the physical resources reserved for it (static costs) and on the utilization made of these resources (dynamic costs). Moreover, the energy costs of a VM are predictable for the static part, and bounded (by the maximum costs as shown in Figures 3 and 4) and assessable through PowerIndex on heterogeneous nodes. Thus the user knows the maximal costs of the VM, and is able to estimate the real costs if the behavior of the running application and their energy consumption is known. With PowerIndex we extend the estimation of the dynamic costs on different machines by offering limited profiling on a reference machine.

However, EPAVE and PowerIndex together are not designed to account for the *real* cost of a given VM as it could be measured by external wattmeters during the entire lifetime of this VM. In this case, the cost of a VM would be influenced by cloud provider operations like VM migration or allocation of other VMs on the same host. This does not seem to be a desirable feature as it would reveal private information from the provider point of view. This is why EPAVE and PowerIndex are not based on this purely measurement technique and also why their goal is not to provide *real* measured costs but predictable, bounded, energy-proportional costs of a VM. These reflect the energy costs of an average VM of a given size hosted on a fixed Cloud platform, similarly to what is done for pricing models [32].

EPAVE provides a complete view of the energy costs related to the hosting of virtual machines. Indeed, it does not only take into account server-related costs, but also the costs of the air conditioning, the networking devices, the power supplies, etc. That is why EPAVE can help the Cloud provider to easily and fairly distribute the energy consumption of its entire infrastructure over the customers.

The computation of the energy costs determined by EPAVE relies, for the static side, on external power measurements (PUE, idle power of the servers), and for the dynamic side, on wattmeters or software-based tools. If these measured information are stored over time, the EPAVE energy costs can be recomputed later, thus becoming verifiable and auditable. PowerIndex only requires offline profiling once for each type of machine, the online profiling relies on wattmeters or software-based estimation. Also values for PowerIndex can be recomputed after the short profiling phase.

EPAVE encourages users to dimension adequately their VMs. PowerIndex additionally helps to motivate the customers to go for the most efficient reservation in terms of energy. With the Utilization Mapping the case of underutilizing the VMs can be avoided. Indeed, if a user is asking for a 4 vCPUs VM, but uses only 2 vCPUs, the two unused vCPUs will still be taken into account into the static costs – although their dynamic costs will be zero, and even if the

Cloud provider is applying over-commitment of resources. Indeed, the dynamic costs are directly measured from the hardware, so all energy saving mechanisms employed by the user (e.g., energy-aware software) will be directly translated into a reduction of the dynamic costs of the VM. We assume here that the energy costs of a VM have somehow repercussions for the user (like a bonus-malus system, or monetary costs for VMs taking into account the energy).

In the case of heterogeneous servers, EPAVE in combination with PowerIndex encourage the Cloud provider to use the most energy-efficient nodes. For instance, for the case described in Figure 4 with the Taurus cluster and the old *Sagittaire* cluster, a VM with 2 vCPUs will have static costs of 105.74 Watts. So, its static costs are bigger than the idle power consumption of a Taurus server, which is still able to host 5 more of such VMs. However, this VM's static costs are nearly twice smaller than the idle power consumption of a *Sagittaire* server which cannot host any additional VM.

6. Outlook

This section gives a non-exhaustive outlook on the application of EPAVE and PowerIndex, and more generically of the utilization of energy-aware cost models.

6.1. Pricing models

EPAVE can serve as the basis for energy-aware pricing models. The static part is known at the beginning as it is defined by the VM type. For the dynamic part, the minimal bound is zero, and the maximal bound (for maximal energy consumption) can be provided to the user before the purchase. Reporting these bounds to the user makes costs per VM predictable (bounded) and keeps the spirit of the pay-as-you-go model because the dynamic part of the costs is in most cases smaller than the static parts (reflecting the reality of the power consumption of typical data center servers).

6.2. SLA with renewable energy sources

Service Level Agreements (SLAs) provide quantified guarantees to the users concerning quality of service on the reserved VMs. In [37], the authors define green SLA: an explicit SLA for the percentage of renewable energy used to run the clients' workloads. In [38], the terms of green SLAs include also the energy costs of networking devices and virtual links between VMs. The green SLA is negotiated between the IaaS provider and each client depending on its needs. Such an SLA requires to have quantifiable green cloud services [37]. That is to say, the provider has to know the energy consumption of each VM and the electricity mix employed by the data centers. EPAVE can be used here to determine the energy budget spent by the VMs of a given user, and thus, to deduce the amount of green energy required for the Cloud provider in order to fulfill the SLA conditions for this user.

6.3. User-oriented utilization

On the user side, EPAVE can be used as an energy cost metric in order to evaluate the energy-efficiency of a given application running on given VM configuration. This metric can be used in combination with the classical metrics (duration, performance, QoS, etc.). By extrapolation, EPAVE can serve as a basis for a cost-benefit analysis including energy costs. Similarly, PowerIndex can be used for comparing different VM configurations for a given application, and thus determine the desirable trade-off between QoS and energy consumption.

Combined with energy-aware pricing models on the Cloud provider side, EPAVE and PowerIndex can be an energy-aware incentive motivation. Energy-efficient users can be rewarded on the basis of their energy cost if they actively act towards its reduction. On the contrary, users can have an energy quota for running their VMs, which can be set by the provider or by the energy-aware user herself.

The application of EPAVE and PowerIndex described here are in particular possible because we do not only consider only the dynamic costs, and therefore, underutilization cases are penalized, as shown in Section 3.3.3. Finally, the

utilization of EPAVE and PowerIndex simply display the energy costs of VMs could help raising energy-awareness of users.

6.4. Open points

EPAVE and PowerIndex leave some questions, which will be the subject of future work. In particular, EPAVE does not account for energy-saving techniques employed by Cloud providers, like switching off idle nodes. Therefore, it cannot be used to measure the energy efficiency of Cloud facilities. Over-commitment is a classical technique employed by IaaS providers in order to decrease resource under-utilization and to maximize profit. EPAVE does not take this into account. **Additionally, the EPAVE cost model presented here mainly focuses on CPU utilization to attribute both dynamic and static costs. This limitation might be overcome in future work by: 1) proposing a cost model based on volume of resources (i.e. depending on the product of required vCPU and memory instead of only on the vCPU quantity) for the static cost, and 2) employing a fine-grain monitoring tool such as BitWatts [5] for the dynamic cost.**

PowerIndex relies on profiling parts of an application to make assumptions on costs on different machines. If the workload is very diverse in terms of power consumption, the prediction might be inaccurate. However, in combination with EPAVE the upper bounds of dynamic costs are known and can lower the risk of misplacement of workloads. This paper presents our first attempt to build a reliable and intuitive model for energy accounting per VM in a heterogeneous Cloud infrastructure. We hope this work will start paving the road towards energy-aware Clouds.

7. Conclusions

In this paper we introduced EPAVE, a model for predictable and transparent energy cost attribution per user. EPAVE is designed for simple usage, trying to keep the effort as limited as possible. The static costs comprise the PUE, which

is already available in many data centers. The remaining static costs only have to be derived once. The only thing that requires constant monitoring are the dynamic costs, whereas the maximum dynamic costs can be pre-determined. In our experiments the actual dynamic costs are measured with a wattmeter as the nodes were used in a single-user mode. For multi-tenant usage a more fine-grained monitoring is required, such as provided by BitWatts [5] that additionally does not require a wattmeter (except for the model building phase). Additionally, BitWatts has been proved to accurately attribute dynamic energy costs among several VMs hosted on the same host for various kinds of workloads [5]. As a help to dimension VMs, we also introduce PowerIndex, a simple profiling tool that allows to display the costs of a workload running on different types of machines. This tool comprises an offline profiling phase required only once. The online monitoring phase is done once per new application, and only performed for a limited amount of time.

Acknowledgments

The work presented in this paper has been partially supported by EU under the COST programme Action IC1305, Network for Sustainable Ultrascale Computing (NESUS).

Experiments presented in this paper were carried out using the Grid'5000 experimental test-bed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see <https://www.grid5000.fr>).

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under the ParaDIME Project (www.paradime-project.eu), grant agreement no 318693.

References

- [1] G. Cook, How clean is your cloud? Report, Greenpeace International, April 2012.

- [2] A.-C. Orgerie, M. Dias de Assunção, L. Lefèvre, A Survey on Techniques for Improving the Energy Efficiency of Large-scale Distributed Systems, *ACM Computing Surveys* 46 (4) (2014) 47:1–47:31. doi:10.1145/2532637.
- [3] A. Bohra, V. Chaudhary, VMeter: Power Modelling for Virtualized Clouds, in: *IEEE International Symposium on Parallel Distributed Processing (IPDPSW)*, 2010, pp. 1–8.
- [4] A. Kansal, F. Zhao, J. Liu, N. Kothari, A. A. Bhattacharya, Virtual Machine Power Metering and Provisioning, in: *ACM Symposium on Cloud Computing (SoCC)*, 2010, pp. 39–50.
- [5] M. Colmant, M. Kurpicz, P. Felber, L. Huertas, R. Rouvoy, A. Sobe, Process-level Power Estimation in VM-based Systems, in: *European Conference on Computer Systems (EuroSys)*, ACM, 2015, p. 14.
- [6] L. A. Barroso, U. Hölzle, The case for energy-proportional computing, *IEEE Computer* 40 (12) (2007) 33–37.
- [7] R. Sen, D. A. Wood, Energy-Proportional Computing: A New Definition, *Computer* 50 (8) (2017) 26–33.
- [8] M. Kurpicz, A. Sobe, et al., How much does a VM cost? Energy-proportional Accounting in VM-based Environments, in: *Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016, pp. 651–658.
- [9] C. Delimitrou, C. Kozyrakis, Paragon: QoS-aware Scheduling for Heterogeneous Datacenters, in: *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, New York, NY, USA, 2013, pp. 77–88.
- [10] J. Mars, L. Tang, R. Hundt, Heterogeneity in "Homogeneous" Warehouse-Scale Computers: A Performance Opportunity, *IEEE Computer Architecture Letters* 10 (2) (2011) 29–32.

- [11] J. Koomey, Growth in Data Center Electricity Use 2005 to 2010, Analytics Press.
- [12] P. X. Gao, A. R. Curtis, B. Wong, S. Keshav, It's Not Easy Being Green, in: ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), 2012, pp. 211–222.
- [13] D. Snowdon, S. Ruocco, G. Heiser, Power Management and Dynamic Voltage Scaling: Myths and Facts, in: Workshop on Power Aware Real-time Computing, 2005, pp. 1–7.
- [14] V. Sivaraman, A. Vishwanath, Z. Zhao, C. Russell, Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router, in: IEEE Conference on Computer Communications Workshops (INFOCOM Workshops), 2011, pp. 331–336.
- [15] E. Pinheiro, R. Bianchini, Energy Conservation Techniques for Disk Array-based Servers, in: Annual International Conference on Supercomputing, ICS '04, 2004, pp. 68–78.
- [16] A.-C. Orgerie, L. Lefèvre, J.-P. Gelas, Demystifying Energy Consumption in Grids and Clouds, in: Work in Progress in Green Computing Workshop (IGCC Workshop), 2010, pp. 335–342.
- [17] How dirty is your data?, Greenpeace report (2011).
- [18] J. Baliga, R. Ayre, K. Hinton, R. Tucker, Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport, Proceedings of the IEEE 99 (1) (2011) 149–167.
- [19] B. Krishnan, H. Amur, A. Gavrilovska, K. Schwan, VM Power Metering: Feasibility and Challenges, ACM SIGMETRICS Performance Evaluation Review 38 (3) (2011) pp. 56–60.
- [20] R. Bertran, Y. Becerra, D. Carrera, V. Beltran, M. González, X. Martorell, N. Navarro, J. Torres, E. Ayguadé, Energy Accounting for Shared Virtual-

ized Environments Under DVFS Using PMC-based Power Models, *Future Generation Computer Systems* 28 (2) (2012) pp. 457–468.

- [21] A. Greenberg, J. Hamilton, D. Maltz, P. Patel, The Cost of a Cloud: Research Problems in Data Center Networks, *ACM SIGCOMM Computer Communication Review* 39 (2008) 68–73.
- [22] M. Pawlish, A. Varde, Free cooling: A paradigm shift in data centers, in: *International Conference on Information and Automation for Sustainability (ICIAFS)*, 2010, pp. 347–352.
- [23] H. Hlavacs, T. Treutner, J.-P. Gelas, L. Lefevre, A.-C. Orgerie, Energy consumption side-channel attack at Virtual Machines in a Cloud, in: *International Conference on Cloud and Green Computing (CGC)*, 2011, pp. 605–612.
- [24] T. Heath, B. Diniz, E. V. Carrera, W. Meira Jr, R. Bianchini, Energy conservation in heterogeneous server clusters, in: *ACM SIGPLAN symposium on Principles and practice of parallel programming*, 2005, pp. 186–195.
- [25] J. Mars, L. Tang, R. Hundt, Heterogeneity in homogeneous warehouse-scale computers: A performance opportunity, *Computer Architecture Letters* 10 (2) (2011) 29–32.
- [26] M. Kurpicz, A. Sobe, P. Felber, Using power measurements as a basis for workload placement in heterogeneous multi-cloud environments, in: *International Workshop on CrossCloud Systems*, ACM, 2014, p. 6.
- [27] C. Delimitrou, C. Kozyrakis, Quality-of-service-aware scheduling in heterogeneous data centers with paragon, *Micro*, IEEE 34 (3) (2014) 17–30.
- [28] R. Nathuji, C. Isci, E. Gorbatoov, Exploiting platform heterogeneity for power efficient data centers, in: *International Conference on Autonomic Computing (ICAC)*, IEEE, 2007, pp. 1–5.

- [29] V. Avelar, D. Azevedo, A. French, PUE: a comprehensive examination of the metric, Green Grid white paper (2013).
- [30] U. Institute, 2014 Data Center Industry Survey, <http://journal.uptimeinstitute.com/2014-data-center-industry-survey/> (2015).
- [31] Natural Resources Defense Council, Data Center Efficiency Assessment, NRDC Issue Paper, <https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IP.pdf> (2014).
- [32] AWS, Amazon EC2 Pricing, https://aws.amazon.com/ec2/pricing/?nc2=h_ls (*accessed in August 2017*).
- [33] Q. Liu, D. Zhang, Dynamic Pricing Competition with Strategic Customers Under Vertical Product Differentiation, *Management Science* 59 (1) (2013) 84–101.
- [34] D. Balouek, A. Carpen Amarie, G. Charrier, F. Desprez, E. Jeannot, E. Jeanvoine, A. Lèbre, D. Margery, N. Niclausse, L. Nussbaum, O. Richard, C. Pérez, F. Quesnel, C. Rohr, L. Sarzyniec, Adding virtualization capabilities to the Grid’5000 testbed, in: I. Ivanov, M. Sinderen, F. Leymann, T. Shan (Eds.), *Cloud Computing and Services Science*, Vol. 367 of *Communications in Computer and Information Science*, Springer International Publishing, 2013, pp. 3–20.
- [35] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, et al., Apache Hadoop Yarn: Yet another resource negotiator, in: *ACM Symposium on Cloud Computing (SoCC)*, 2013, p. 5.
- [36] S. Huang, J. Huang, J. Dai, T. Xie, B. Huang, The HiBench benchmark suite: Characterization of the MapReduce-based data analysis, in: *IEEE Data Engineering Workshops (ICDEW)*, 2010, pp. 41–51.

- [37] M. Haque, K. Le, I. Goiri, R. Bianchini, T. Nguyen, Providing green SLAs in High Performance Computing clouds, in: International Green Computing Conference (IGCC), 2013, pp. 1–11.
- [38] A. Amokrane, M. Zhani, Q. Zhang, R. Langar, R. Boutaba, G. Pujolle, On satisfying green SLAs in distributed clouds, in: International Conference on Network and Service Management (CNSM), 2014, pp. 64–72.