Virtual Machine Power Accounting with Shapley Value

Weixiang Jiang[1], Fangming Liu*[1], Guoming Tang[2], Kui Wu[2], and Hai Jin[1]
[1]Key Laboratory of Services Computing Technology and System, Ministry of Education,
School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2]University of Victoria, Canada

***Abstract*— The ever-increasing power consumption of datacenters has eaten up a large portion of their profit. One possible solution is to charge datacenter users for their actual power usage. However, it poses a great technical challenge as the power of VMs co-existing in a physical machine cannot be measured directly. It is thus critical to develop a fair method to disaggregate the power of a physical machine to individual VMs. We tackle the above challenge by modeling the power disaggregation problem as a cooperative game and propose non-deterministic Shapley value to discover the *fair* power share of VMs (in the sense of satisfying four desired axiomatic principles), while compensating the negative impact of VM power variation. We demonstrate that the results from existing power model-based solution can deviate from the "ground truth" by $25.22\% \sim 46.15\%$. And compared with the exact Shapley value, our non-deterministic Shapley value can achieve less than 5% error for 90% of the time.**

I. Introduction

With the rapid development of cloud computing, datacenters as the core infrastructure of cloud computing have been deployed all over the world. As a result, datacenters have become the largest and fastest-growing population of electricity consumers [1]. While tremendous efforts have been made to tackle the increasing power expenses, such as using server consolidation and hardware over-provisioning to reduce the energy cost [2], [3], [4], [5], [6], [7], datacenter operators are often faced with the headache that datacenter tenants should be charged fairly to cover large energy bills.

With the expanding datacenter scale and the increasing electricity price, the datacenter operating expenditure (OPEX) has become much higher [8]. Table I illustrates the price comparison between the costs of electricity and IT resources, showing that the electricity cost is chasing the IT hardware cost. In addition, electricity cost builds up quickly over the datacenter running time. Under such circumstances, it is reasonable to consider the energy consumption of datacenter tenants in the datacenter pricing scheme.

Current VM pricing policy is simply based on the type of VM and time of use, such as Amazon EC2 instance pricing [11] and Microsoft Azure pricing [12]. Nevertheless, such policies can cause severe unfair charging in practice. As an example shown in Fig. 1, user A and user B rent the same type of VMs in the same time period $[T_0, T_5]$. During the VMs usage, however, their workloads stress the VMs at different resource (e.g., CPU) utilization levels, which leads to different power consumptions. Consequently, user B actually consumes 33% more energy than user A in total, whereas they pay the same bill based on the current pricing model. Therefore, a fairer VM pricing model that relies on its exact energy consumption is highly demanded, especially in the virtualized datacenters with multiple tenants. In addition, similar energy pricing for servers has already been applied in co-location datacenter [13], [14], [15].

TABLE I
Comparison of different resource costs to support a mid-level configured VM in AWS per year

Instance Type	*Electricity Cost*		*CPU*	*RAM*	*SSD*
	USA	*Germany*	*Cost*	*Cost*	*Cost*
General Purpose	\$100.74	\$193.52	\$310.4	\$80	\$26
Computed Optimized	\$105.15	\$201.94	\$349	\$40	\$26
Memory Optimized	\$100.74	\$193.52	\$310.4	\$160	\$26
Storage Optimized	\$100.74	\$193.52	\$310.4	\$160	\$256

[1] The update cycle of IT hardware is 5 years.
[2] A middle-level VM in AWS is configured with 16 vCPUs, which is equal to an Intel Xeon CPU. The electricity cost is calculated with the designed power of corresponding Xeon CPU and electricity price in 2015 (not including other components and power infrastructure cost) [9], [10].

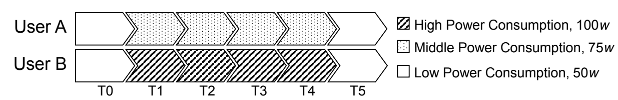


Fig. 1. Different power usage patterns of two users on the same VMs.

Power measurement of VM has a far-reaching importance much beyond fair energy pricing. Understanding the power patterns of VM helps better VM provisioning and workload scheduling in the datacenter [16], [17], [18], [19]. VM power measurement is also critical to the overall datacenter power management. For instance, to reduce power provisioning costs, VM power measurement can effectively enable power caps to be enforced on a per-VM basis.

While important, the power measurement of a VM is technically difficult, as we cannot attach any power meters to a VM. Recently, solutions based on power model to VM power estimation have been developed [20], [21], [22], [23]. As a software-based solution, the power model-based approach,

*This work was supported in part by NSFC under Grant 61520106005, in part by the National Key Research & Development (R&D) Plan under grant 2016YFB1000500, in part by the National 973 Basic Research Program under Grant 2014CB347800, and in part by the Natural Sciences and Engineering Research Council of Canada under the grant (No. 195819339). (Corresponding author: Fangming Liu)

which links the power consumption of a server with its performance metrics exposed by OS (e.g., CPU utilization), has been widely adopted for power estimation of individual physical servers [24], [25], [26]. Nevertheless, the same idea may not be effective in the virtualized environment. As we will show later (Fig. 4 in Sec. III), the power model-based approach, when applied for VM power estimation, can lead to a discrepancy, up to 25.22% on Pentium CPU and 46.15% on Xeon CPU, between the estimate and "ground truth". This is because the power model fails to include the correlation of different VMs' power consumption. In the virtualized environment, multiple VMs are accommodated in the same physical machine and share the same hardware. Hence, as will be analyzed in Sec. III, the VMs actually compete for hardware resources (e.g., computing units, registers and caches), and thus can affect each others' power consumption.

With the above observation, we are motivated to develop a fair solution that disaggregates the overall power of a physical machine into the power of each VM. Intuitively, the power disaggregation problem is similar to the surplus allocation problem in cooperative game: how much does each player contribute to the overall cooperation and what payoff should they receive in the cooperative game? Following this intuition, we propose a new approach that makes use of cooperative game theory to estimate the power of individual VMs residing in the same physical machine.

Specifically, we make the following major contributions in this paper:

- We apply the Shapley value in cooperative game to estimate the power consumption of individual VMs. This approach is a natural adaptation to the interactions among different VM coalitions. It provides a fair VM power allocation that can be used for VM power management and pricing.
- We extend the original Shapley value to non-deterministic Shapley value. The extended solution and its corresponding heuristic can solve the VM power disaggregation problem effectively with a much-reduced complexity.
- We implement our method and thoroughly test its performance over various benchmark applications. The results show that compared with the exact Shapley value, our non-deterministic Shapley value can achieve less than 5% error for 90% of the time.

The rest of the paper is organized as follows: In Sec. II, we review related works on VM power estimation. We conduct an experiment to reveal the phenomenon of VM power interaction in Sec. III. The background of Shapley value and its advantages in solving the VM power estimation problem are introduced in Sec. IV. In Sec. V, we define the non-deterministic Shapley value and present its solution. We implement our method in Sec. VI and evaluate its performance in Sec. VII. Sec. VIII discusses the limitations and open challenges of this work. The paper is concluded in Sec. IX.

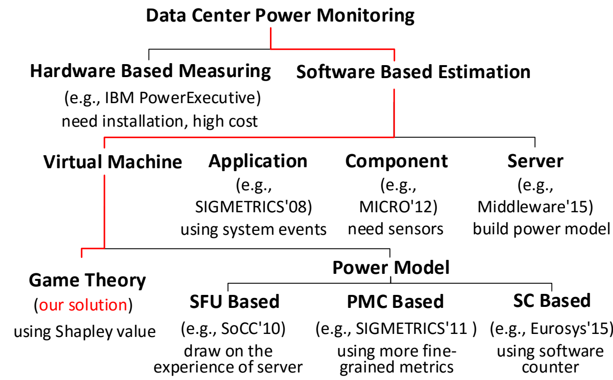


Fig. 2. Classification of power measurements at different levels in the datacenter.

II. Related Work

A. Power Model for Server and Component

Intel's Running Average Power Limit (RAPL) [27] exposes a set of CPU counters in order to provide energy and power consumption information for Intel Sandy Bridge and Ivy Bridge CPUs. RAPL is not an analog power meter, but rather use a software power model. Based on RAPL, Zai et al. [28] introduce a hyperthread-aware power model, which dynamically estimates the power of individual processes or threads. To access more fine-grained hardware power information of CPU, Huang et al. [29] construct power proxies for CPU, at both chip level and core level, by utilizing specialized activity counters in the chip hardware. To reduce hardware cost, a software based power disaggregation approach is proposed in [24] to estimate the power consumption of individual servers in legacy datacenters.

B. VM Power Estimation

Solutions Based on System Functional Unit (SFU): Similar to server-level power estimation, power model is also adopted to estimate VM power. In [20], Joulemeter is initially established by building a linear power model for VM based on system functional units (SFU), including the usage of CPU, memory, disk and network card. Then the usage of each isolated SFU is mapped to the value in the power model. However, the accuracy of SFU-based power model suffers from the unobserved states of CPU, such as CPU I/O wait caused by cache competition among VMs [20], [21].

Solutions Based on Performance Metric Counter (PMC): To improve the accuracy of VM power estimation, performance metric counters (PMCs) are utilized. In [21], a power metering method for CPU intensive and RAM intensive VMs is proposed by using more specific PMCs, including instructions retired per second and last-level cache misses per second. Along the line, VMeter [22] makes use of four PMCs to monitor system sub-components exposed by AMD Opteron processor. Different from previous works, VMeter does not map isolated PMC to the VM power consumption, but takes into account the correlation between the pairs $\{$CPU, cache$\}$

and {disk, DRAM} and decomposes the power model into two linear regression models.

Solutions Based on Software Counter (SC): In addition to the power model based on hardware PMCs, SC-based power models are also developed to estimate the power in virtualized system. WattApp [30] builds application-aware power model with application throughput instead of PMCs. Motivated by nested virtualization, BITWATTS [23] develops a process-level power model to account the power of each process, and then the process-level power is mapped to nested VMs.

C. Solutions Using Shapley Value

The Shapley value in cooperative game theory has been applied to solve problems in shared computer systems [31], [32], [33]. Tracking application energy usage in mobile system [33] is mostly relevant to our work. Nevertheless, there are three significant differences between the work in [33] and ours. First, the work in [33] focuses on accounting accumulated energy for applications in the mobile system, while our work deals with real-time power metering of VMs in virtualized environment. Second, VM is more complicated than applications, e.g., different applications/jobs running in a VM. Finally, in [33], the aggregated energy consumption is investigated as a non-deterministic variable related to application-independent factors (e.g., LCD brightness), but the energy variation caused by different resource usage of an application is out of consideration. In contrast, VM power varies a lot in different resource usage states, making the solution of Shapley value more sophisticated than that in [33].

In summary, as illustrated in Fig. 2, our solution to the VM power estimation is unique in the space of datacenter power monitoring[34], [35], [36], [37].

III. Understanding the VM Power

In this section, we conduct experiments to reveal the characteristics of VM power interaction. Then, we analyze the impact of modern computer system architecture on the VM power estimation.

A. Experimental Setup

Measurement platform: Two physical machines with Intel Pentium and Intel Xeon CPU, respectively, are used in our measurement. We choose the Intel's X86 architecture due to its wide adoption in virtualization platforms, such as the major cloud service provider Amazon [9]. In addition, we use a power meter to record the physical machine's real-time power consumption with frequency of 1 Hz. Since it is impossible to measure individual VM's power directly when several VMs co-exist in a physical machine, we first refer to the prior work based on VM power model. In [20], [21], [22], [23], to train the power model of a VM, the marginal power contribution of the VM, which refers to the power increase/decrease after starting/shutting down the VM while keeping other VMs idle, is used as the "ground truth" of the VM power. We implement the same method for comparison.

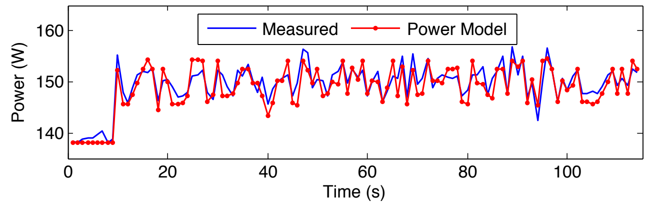
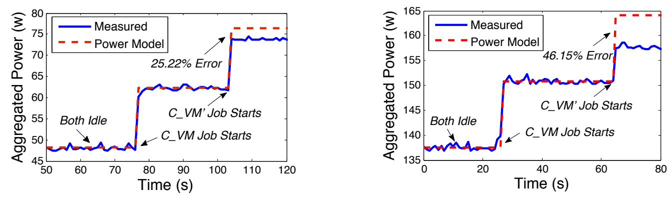


Fig. 3. Power estimation of the physical machine using integrated VM power model.



(a) Pentium (b) Xeon

Fig. 4. Power estimation using independent VM power model.

VM Setup: Two identical VMs (C_VM and C_VM') are created on the same physical machine, both with configuration of 1 vCPU, 512 MB memory, 8 GB disk and Linux OS. In this demonstration, we target at the CPU (which is the most power-hungry component) to reveal the resource competition among VMs as well as its impact on power consumption.

B. Power Model of Integrated VMs

We program a synthetic benchmark which randomly consumes CPU cycles and run it on both Xeon CPU-based C_VM and C_VM' simultaneously. Treating both VMs as a whole, we train a power model for the whole physical machine using the collected VM states and corresponding power consumptions of physical machine:

$$p' = 9.49u' + 138, \tag{1}$$

where p' is the power of the physical machine, u' is the sum of CPU utilization of C_VM and C_VM' and 138 is the idle power[1] of physical machine. Based on the power model, we can estimate the power consumption of the physical machine using VMs' total CPU utilization, as shown in Fig. 3.

We can see that the power model performs well at the level of whole system (with an average relative error of 2.07%). This observation is consistent with the results in [20]. Next, we disclose the pitfalls when the power model is applied for individual VM power estimation.

C. Power Model of Individual VM

We perform the floating-point calculation job (*"scale=6000; 4*a(1)"* | *bc -l -q*) on both VMs, with which we can stress the CPU utilization close to 100% while keeping other components nearly idle. According to power model, C_VM and C_VM' should contribute the same power to the physical

[1]The physical machine's idle power is stable, here we take it as a constant.

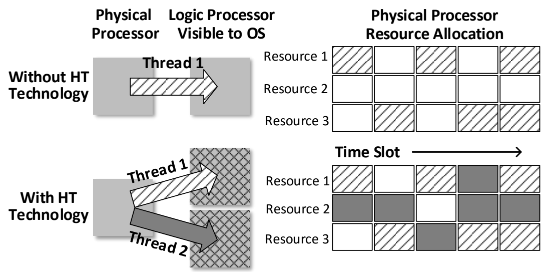


Fig. 5. An example showing how hyper-threading technology works.

machine given the same state (i.e., 100% CPU utilization). We keep the hypervisor idle during the job execution and only start C_VM and C_VM'. By activating the jobs on the VMs, we measure the power consumption of physical machine. The measurement results from the Pentium CPU and from the Xeon CPU are shown in Fig. 4 (a) and (b), respectively.

Take the Xeon CPU as an example for detailed analysis. We can see that when both VMs keep idle, the power of physical machine is about 138 W. When we activate the floating-point calculation job on C_VM, the power of the physical machine increases to around 151 W.

Following the training process in previous section (which was also adopted in [20], [21], [22], [23]), the VM power model for C_VM or C_VM' should be:

$$p = 13u \tag{2}$$

where p is the power of VM and u is its CPU utilization.

Then, we verify the accuracy of VM power model when the jobs are activated on both VMs in sequence. First, we start the job on C_VM, and the power estimation of the power model is accurate (with an average relative error of 0.23%). When we continue to start the job on C_VM', however, it results in a relative error of 46.15%! According to the power model, C_VM' should contribute 13 W to the whole system while the measured value is only 7 W. We change the job execution order of C_VM' and C_VM, while observing the same phenomenon (i.e., the first active VM brings 13 W, the second active VM brings 7 W). In addition, such an error can be several times larger if the VMs are configured with a higher number of vCPUs, such as *c3.4xlarge* instance with 16 vCPUs in Amazon EC2 [9].

D. How do the Errors Occur?

We take a closer look at the architecture of the modern processor, which adopts the hyper-threading technology (HT-T). A physical core with HTT consists of two logical cores, each of which can execute a specified work while sharing the same physical core. Fig. 5 shows a detailed example: when two threads running on the same core, HTT helps fill tasks into the idle resources to improve the efficiency of CPU.

Nevertheless, this introduces competition in our context when two VMs request a limited number of resource at the same time (e.g., float computing unit or float register), and consequently only one VM can possess the conflicted resource.

TABLE II
NOTATIONS

Symbol	Description
$\mathcal{N}$	The set of n VMs that need to be estimated
$\mathcal{S}$	A subset of $\mathcal{N}$; a coalition of VMs
k	Number of a VM's component states
$\mathbf{c}_i$	The component state vector of VM i
$\mathcal{C}$	The component state vector set of a coalition of VMs
$\mathbf{v}_j$	The sum of $\mathbf{c}_i$ in the same VHC j
$\mathbf{w}_j$	Power mapping vector for VHC j
Φ_i	Power of VM i
$v(\mathcal{S})$	Power of coalition $\mathcal{S}$
$\Phi_i(\mathcal{C})$	Power of VM i in a coalition with state $\mathcal{C}$
$v(\mathcal{S},\mathcal{C})$	Power of coalition $\mathcal{S}$ with state $\mathcal{C}$
VHC	Virtual Homogeneous VM Coalition

Thus, at any time instant, only one VM is actually using a specific unit of the physical core instead of two. However, at the view of OS level, it seems that both VMs are using the CPU with 100% utilization. As a result, the aggregated power estimation based on VM power model is higher than the power of the physical machine. The power model needs to be rebuilt. Unfortunately, such detailed PMC of HTT is not exposed to the OS and the actual power of individual VMs co-existing in the same physical machine can not be obtained. Without the detailed PMC and ground truth, it is impossible to build accurate power models for individual VMs. Thus, a different approach to VM power estimation is needed.

Remark 1: There always exists idle power consumption for a virtualized physical machine and it does not change after adding any idle VMs[2], which means an idle VM does not bring any extra power consumption to the physical machine. So in the rest of the paper, when we compare the power of a physical machine with the aggregated power of VMs, the power of a physical machine always refers to the adjusted power, which deducts the idle power of the physical machine. How to attribute the idle power to VMs will be discussed in Section VIII.

IV. Why Shapley Value?

In this section, we first formally define the problem of VM power estimation. Then, we overview the background of Shapley value and explain why Shapley value can be applicable to the VM power estimation problem. For ease of reference, Table II lists the notations used in later sections.

A. Problem Definition

Consider a set of VMs (denoted as $\mathcal{N} = \{1, 2, ..., n\}$) running on the physical machine at a given time. The objective of VM power estimation is to determine the power contribution of VM i (denoted as $\Phi_i, i \in \mathcal{N}$) to the physical system power P. Mathematically a power estimation method provides a way

[2]While creating a VM, the physical machine's power will increase. But once the creation process is finished, the physical machine's power goes back to the same as before.

TABLE III
AN EXAMPLE TO COMPARE DIFFERENT POWER ALLOCATION MECHANISMS FOR TWO IDENTICAL VMS IN FIG. 4(B)

Allocation Mechanism	C_VM	C_VM'	$C\_VM + C\_VM'$	*Measured Power*	Macro-level Accuracy	*Fairness*
Marginal Contribution	13 W	7 W	20 W	20 W	✓	×
Power Model	13 W	13 W	26 W	20 W	×	✓
Ideal Estimation	10 W	10 W	20 W	20 W	✓	✓

of mapping from the aggregated power P (i.e., the measured power of the physical machine) to per-VM power Φ_i.

The ideal VM power estimation is expected to satisfy the following requirement: the estimated Φ_i should be equal to the actual power consumption of VM i, which we named as *micro-level accuracy*. Nevertheless, as VMs co-exist in the same physical machine and interact with each other, it is impossible for us to obtain the actual power of individual VMs, nor can we validate the micro-level accuracy requirement. Thus, instead of micro-level accuracy, we focus on the *macro-level accuracy* and *fairness* of the VM power estimation:

1) **Macro-level Accuracy**: The aggregated power estimations of all VMs should be equal to the measured power of physical machine, i.e., $\sum_{i \in \mathcal{N}} \Phi_i = P$.
2) **Fairness**: The power allocations among all VMs, i.e., Φ_i, should be fair enough.

To further clarify the concepts of macro-level accuracy and fairness, we use the power estimation of the two VMs (C_VM & C_VM') in Fig. 4(b) as an example. Table III shows the performance comparison of different power estimation strategies.

- Power estimation based on the marginal contribution: As we can see, the sum of power estimations provided by this strategy is equal to the measured power of physical machine, which satisfies the macro-level accuracy requirement. Nevertheless, it gives different power allocations to the two identical VMs (i.e., the same configuration, executing the same job and consuming the same CPU resource), which we claim is against the fairness requirement.
- Power estimation based on the power model: Following the power model given by Equation 2, the two identical VMs will be allocated exactly the same power, which is fair enough; whereas the sum of each VM's power allocation (13 W) exceeds the actual power measurement of physical machine (20 W), which violates the macro-level accuracy.

Here, we can guess that the ideal power allocation for either VM should be 10 W, such that both macro-level accuracy and fairness could be achieved. Rather than guessing the power, is there any new power estimation scheme that we can apply, so that both macro-level accuracy and fairness can be guaranteed? Our solution is Shapley value. As to why choose Shapley value and how it fits for VM power estimation, we will introduce them in the following sections.

B. Background on Shapley Value

The two requirements in the problem of VM power estimation resemble the conditions in a cooperative game, where it is required to fairly distribute among the players the total revenue/cost generated by the coalition of all players. In our context, each VM is a player, and co-existing VMs on the same physical machine form a cooperative game. The measured power of physical machine represents the total "revenue", generated by the co-existing VMs. The goal is to fairly distribute the total "revenue" among the co-existing VMs.

Shapley value [38] was proposed to solve the above problem. Applying the terminology of [38] in our context, we call a subset $\mathcal{S} \subseteq \mathcal{N}$ a coalition of VMs. For each coalition $\mathcal{S}$, $v(\mathcal{S})$ is the *worth function*, which denotes the aggregated power of coalition $\mathcal{S}$.

Shapley value in a cooperative game is uniquely defined with the following four axioms:

Axiom 1: (Efficiency) It requires that the sum of the player's payoff equals the revenue, i.e.,

$$\sum_{i \in \mathcal{N}} \Phi_i = v(\mathcal{N}). \tag{3}$$

Axiom 2: (Symmetry) Player $i, j \in \mathcal{N}$ are symmetric in a game if they make the same marginal contribution to any coalitions. Symmetric players are paid equal shares, i.e., if for each $\mathcal{S} \subseteq \mathcal{N}$ with $i, j \notin \mathcal{S}$, $v(\mathcal{S} \cup \{i\}) = v(\mathcal{S} \cup \{j\})$, then $\Phi_i = \Phi_j$.

Axiom 3: (Dummy) Zero payoffs would be allocated to the players whose marginal contribution is null with respect to every coalition, i.e., if $v(\mathcal{S} \cup \{i\}) - v(\mathcal{S}) = 0$ for every $\mathcal{S} \backslash \{i\} \subseteq \mathcal{N}$, then $\Phi_i = 0$.

Axiom 4: (Additivity) If we combine two individual games into one, then the sum of payoffs ($\Phi_i^{'}$ and $\Phi_i^{''}$) allocated to a player in these two individual games should be equal to the payoff (Φ_i) the player receives from the combined one, i.e., $\forall i$, $\Phi_i^{'} + \Phi_i^{''} = \Phi_i$.

The Shapley value that provably and strictly satisfies the above axioms [38] can be computed as follows:

$$\forall i \in \mathcal{N}, \Phi_i = \sum_{\mathcal{S} \subseteq \mathcal{N} \backslash \{i\}} \frac{v(\mathcal{S} \cup \{i\}) - v(\mathcal{S})}{(|\mathcal{N}| - |\mathcal{S}|)\binom{|\mathcal{N}|}{|\mathcal{S}|}}, \tag{4}$$

where $|\mathcal{N}|$ and $|\mathcal{S}|$ are the cardinalities of $\mathcal{N}$ and $\mathcal{S}$, respectively, and $\binom{|\mathcal{N}|}{|\mathcal{S}|}$ refers to the combination of choosing $|\mathcal{S}|$ from collection $\mathcal{N}$ without repetition.

What does Shapley value explain? We still use the C_VM and C_VM' in Fig. 4(b) as an example, where $\mathcal{N} = \{C\_VM, C\_VM'\}$ and $|\mathcal{N}| = 2$. Fig. 6 shows the marginal

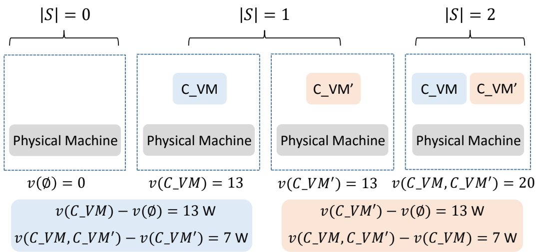


Fig. 6. The marginal power contributions of two VMs in Fig. 4(b).

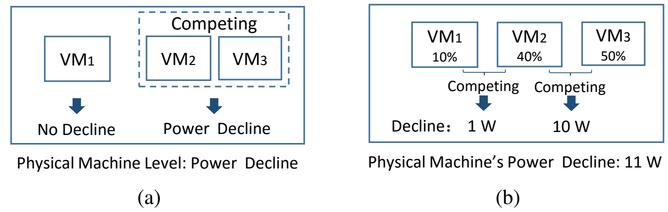


Fig. 7. Fairness analysis.

power contribution of C_VM and C_VM' to different coalitions. According to Shapley value, we have

$$\Phi_{C\_VM} = \frac{13}{(2-0)*\binom{2}{0}} + \frac{7}{(2-1)*\binom{2}{1}} = 10\ W,$$

$$\Phi_{C\_VM'} = \frac{13}{(2-0)*\binom{2}{0}} + \frac{7}{(2-1)*\binom{2}{1}} = 10\ W$$

We can see that Shapley value fairly allocates the physical machine's power to each VM, which results in consistent solution to the ideal estimation in Table III. In fact, $v(\mathcal{S}\cup\{i\})-v(\mathcal{S})$ is the marginal contribution of player i to a coalition $\mathcal{S}$. In other words, Shapley value is a distribution rule based on players' marginal contributions.

Can we just rescale the power of physical machine based on each VM's resource usage (named as *resource usage-based allocation*), and get the same solution as that from Shapley value? In fact, these two strategies are essentially different, and the resource usage-based allocation can lead to unfair power estimations under some situations. Fig. 7 shows two different scenarios of VMs competition. In Fig. 7(a), VM_2 and VM_3 competing for resources results in some power decline, which is eventually reflected by the physical machine's power. If we allocate the physical machine's power based on resource usage, VM_1 will also get a power decline, even though it actually makes no contribution. Similarly in Fig. 7(b), VM_1 gets more power decline (i.e., 1.1 W) based on resource usage-based allocation, while its competition with VM_2 only brings 1 W power decline. These competitions make the resource usage-based allocation unfair. Shapley value, however, is based on marginal power contribution to all the possible subsets. Thus, it can capture that VM_1 joining the coalition (VM_2 and VM_3) does not bring any power decline (in Fig. 7(a)) or brings only 1 W power decline (in Fig. 7(b)). As a conclusion, the Shapley value based solution is fairer than resource usage-based allocation.

C. Advantages of Shapley Value

By treating VMs as players in a cooperative game, we discuss the implication of Shapley value's four axioms for VM power estimation.

Efficiency states that the aggregate of all VMs' estimated power is equal to the measured power of a physical system. As shown in Fig. 4, power model reports a larger power estimation value than the measured one. Pricing with VM power estimated by power models may result in overcharging to users.

Symmetry guarantees two VMs should be allocated to the equal power if replacing one with the other does not change the power in any coalitions. It is obvious that C_VM and C_VM' in Sec. III are symmetric given the same status. Nevertheless, the power of C_VM and C_VM' are different according to marginal power contribution as shown in Fig. 4. On the other hand, the power model is trained with the marginal power contribution. Choosing different marginal power contributions as the training dataset results in different power models for the same VM and may violate *Symmetry*.

Dummy requires that zero power should be allocated to a VM if adding the VM to the physical system does not change the power. According to our experiments, the measured powers with and without the idle VM are nearly identical. This is mainly because when the VM becomes idle, the hypervisor might only keep the OS image in memory without allocating any actual hardware resources to the VM. Nevertheless, with the power model-based estimation, the VM always gets a positive power contribution.

Additivity is an important property of Shapley value. In practice, a VM running on a compute server can be assigned a logic disk hosted by another server (e.g., disk array), to extend the VM's storage space. By applying Shapley value, however, we can treat such a VM in two games and compute the power of the two parts separately. According to *additivity* of Shapley value, the aggregated power of these two parts is the VM's total power.

Accepting the above axioms as the necessary properties in VM power estimation, we conclude that the Shapley value is more suitable than the power model-based approach for VM power estimation.

V. Shapley Value-Based Power Estimation

In this section, we first define the non-deterministic Shapley value-based power estimation. Then we develop a fast linear approximation to compensate the negative impact of VM power variation incurred in applying Shapley value.

A. Non-deterministic Shapley Value

While Shapley value has provided an intuitive solution to VM power estimation, its direct application turns out to be non-trivial, since the power contribution of a VM may vary due to its resource consumption status. For example, the power of a VM highly depends on its CPU utilization and may change over time. As a result, $v(\mathcal{S})$ should be treated as a non-deterministic variable. We thus need to develop a solution to non-deterministic Shapley value.

Assume that n VMs run in a physical machine at a given time. A VM $i, i \in \mathcal{N} = \{1, 2, ..., n\}$ is denoted by its component states[3] using a state vector $\mathbf{c}_i$:

$$\mathbf{c}_i := [c_i^1, c_i^2, ..., c_i^k], \tag{5}$$

where k is the number of components whose state is available.

For each coalition $\mathcal{S} \subseteq \mathcal{N}$, we extend Φ_i to $\Phi_i(\mathcal{C})$ and $v(\mathcal{S})$ to $v(\mathcal{S}, \mathcal{C})$ by considering different states of VMs, where $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{|\mathcal{S}|}\}$ is a set of VMs' states in coalition $\mathcal{S}$. Accordingly, we denote the aggregated power of coalition $\mathcal{S}$ as:

$$v(\mathcal{S}, \mathcal{C}) := v(\mathcal{S}, \mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{|\mathcal{S}|}). \tag{6}$$

Definition 1: Given a VM set $\mathcal{N}$ running on a physical machine with corresponding state $\mathcal{C}' = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{|\mathcal{N}|}\}$ and physical machine's power $v(\mathcal{N}, \mathcal{C}')$, non-deterministic Shapley value-based power estimation is to disaggregate $v(\mathcal{N}, \mathcal{C}')$ to each VM's power $\Phi_i(\mathcal{C}')$ by:

$$\Phi_i(\mathcal{C}') = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}, \mathcal{C} \subseteq \mathcal{C}' \setminus \{\mathbf{c}_i\}} \frac{v(\mathcal{S} \cup \{i\}, \mathcal{C} \cup \{\mathbf{c}_i\}) - v(\mathcal{S}, \mathcal{C})}{(|\mathcal{N}| - |\mathcal{S}|)\binom{|\mathcal{N}|}{|\mathcal{S}|}}. \tag{7}$$

B. Complexity Analysis

Obviously there are 2^n subsets of a VM set $\mathcal{N}$. So the difficulty of applying Shapley value lies in two aspects: (i) Calculating Shapley value requires 2^n operations, which may bring very high overheads; (ii) it requests 2^n $v(\mathcal{S}, \mathcal{C})$ as the inputs.

Theoretically there is no limitation on the number of VMs running on a physical machine concurrently, as long as the physical machine has enough resources. In practice, however, the number of running VMs that a physical machine can host is determined by the number of logical cores (vCPU) for performance consideration. In large datacenters, Intel Xeon series CPUs are widely implemented (e.g., Amazon EC2 [9]). For the lowest spec machine with 16-core, the minimum vCPU number of VMs is 1, and for the highest spec machine with 32-core, the minimum vCPU number of VMs is 2. So in real-world datacenter like Amazon EC2, there are at most 16 VMs that can run on the same physical machine concurrently. The overhead of calculating Shapley value is very low (no more than 2^n operations with $n \leq 16$, i.e., $2^n \leq 65536$).

Though the calculation of Shapley value is easy, collecting the input information (i.e., the measurements for 2^n $v(\mathcal{S}, \mathcal{C})$s) is tedious and time-consuming. For example, to collect $v(\mathcal{S}, \mathcal{C})$ for 16 VMs, we need to run all VMs and traverse 2^{16} subsets. It is infeasible to obtain all of the $v(\mathcal{S}, \mathcal{C})$ by measurements, let alone considering different VM states. So the key to our Shapley value-based VM power estimation is to obtain the $v(\mathcal{S}, \mathcal{C})$s.

[3]The component states of a VM refer to the CPU utilization, memory usage, disk I/O and so on.

C. VHC-based Linear Approximation for $v(\mathcal{S}, \mathcal{C})$

Note that $v(\mathcal{S}, \mathcal{C}))$ means the aggregated power of VMs, which is the power of physical machine (deducting the idle power) with coalition $\mathcal{S}$ running on it. So the problem can be transformed into how to obtain the physical machine's power with different VM coalitions running on it. Motivated by the observation in Sec. III-B, we propose a VHC-based linear approximation for $v(\mathcal{S}, \mathcal{C})$.

1) VHC: It is important to note that in a datacenter VMs are configured with fixed resources. For example, each VM instance type only provides no more than 5 fixed configuration options in Amazon EC2, so the type of VMs running on a physical machine is limited. Therefore, we logically divide the VMs in a coalition $\mathcal{S}$ into virtual homogeneous VM coalitions (VHCs), so that in each VHC, the configuration and hardware of the VMs are the same.

For r types of VMs, we can divide them into r VHCs. As we have introduced, the number of VM types on a physical machine is small, so we can easily traverse 2^r combinations, such as $2^5 = 32$.

We further define the $v(\mathcal{S}, \mathcal{C})$ for any coalition $\mathcal{S}$ as:

$$v(\mathcal{S}, \mathcal{C}) := v(\mathcal{S}, \mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{|\mathcal{S}|}) := v(\mathcal{S}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r), \tag{8}$$

where $\mathbf{v}_j = \sum \mathbf{c}_i, (j = 1, 2, \cdots, r)$ is the sum of VM state vectors in VHC j, and the $\sum$ means vector addition.

Then, the task of measuring the power of 2^n VM combination is simplified into measuring $v(\mathcal{S}, \mathcal{C})$ by traversing 2^r combinations of VHCs. However, for each combination of VHCs, traversing all the combinations of the state $\mathcal{C} := \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r\}$ is still infeasible. For example, considering the CPU utilization which varies from 0.01 to 1, there are 100^5 state combinations for a coalition with 5 VHCs.

Next we introduce a linear estimation to approximate unobserved $v(\mathcal{S}, \mathcal{C})$, which makes use of partially measured $v(\mathcal{S}, \mathcal{C})$s that have the same VHCs to the unobserved state.

2) Linear Estimation for $v(\mathcal{S}, \mathcal{C})$: For each VHC j in coalition $\mathcal{S}$, we define a linear power mapping vector $\mathbf{w}_j = (w_j^1, w_j^2, ...w_j^k)$ that maps the state $\mathcal{C} := \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r\}$ to the power of coalition $\mathcal{S}$ by:

$$v(\mathcal{S}, \mathcal{C}) = \sum_{j=1}^{r} \mathbf{w}_j \mathbf{v}_j. \tag{9}$$

Given the measured powers of m VM coalitions which are denoted as $\mathcal{V} = \{v(\mathcal{S}_1, \mathcal{C}_1), v(\mathcal{S}_2, \mathcal{C}_2), \cdots, v(\mathcal{S}_m, \mathcal{C}_m)\}$, each coalition having the same r types of VMs, but with different running states, the following least-square optimization is applied to find the optimal estimation for the unknown power $v(\mathcal{S}_x, \mathcal{C}_x)$:

$$\text{minimize} \sum_{v(\mathcal{S}, \mathcal{C}) \in \mathcal{V}, \mathbf{v}_j \in \mathcal{C}} ||v(\mathcal{S}, \mathcal{C}) - \textstyle\sum_{j=1}^{r} \mathbf{w}_j \mathbf{v}_j||,$$

which results in the closed form solution:

$$v(\mathcal{S}_x, \mathcal{C}_x) = \sum_{j=1, \mathbf{v}_j \in \mathcal{C}_x}^{r} \mathbf{w}_j \mathbf{v}_j. \tag{10}$$

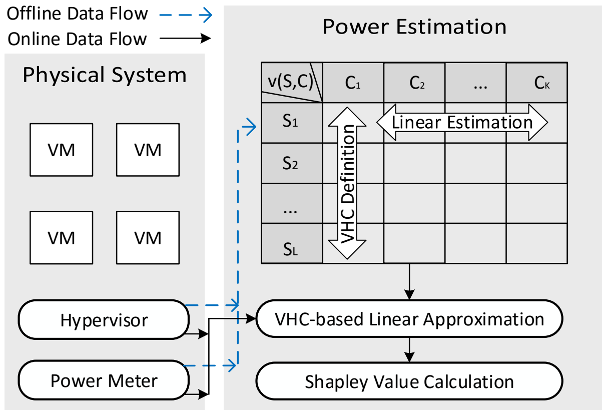


Fig. 8. The framework of Shapley value-based VM power estimation.

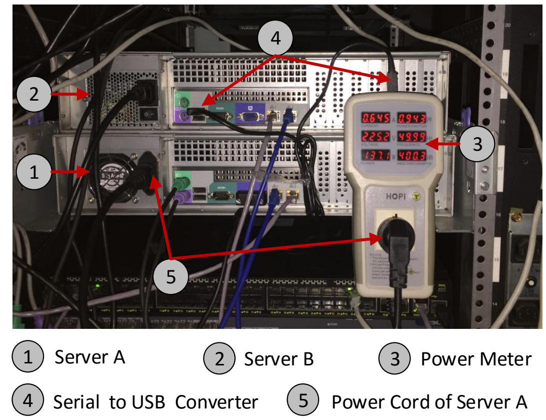


Fig. 9. Prototype: Two servers and one power meter.

Then we can easily obtain any $v(\mathcal{S}, \mathcal{C})$ with partially measured $v(\mathcal{S}, \mathcal{C})$ and calculate each VM power by substituting the corresponding $v(\mathcal{S}, \mathcal{C})$ into Equation (7).

Definition 2: In any coalition $\mathcal{S}$ consisting of r VHCs, VHC-based linear approximation is to determine a set of vectors $\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_r\}$ which map the states of different VHCs in coalition $\mathcal{S}$ to the physical machine's power (i.e., the aggregated power of VMs) when coalition $\mathcal{S}$ running on it.

VI. Implementation

In this section, we introduce our framework of Shapley value-based power estimation. And based on it, we build a real-world prototype. Meanwhile, we also introduce how we collect VMs' states and physical machine's power.

A. Framework

The framework of our Shapley value-based power estimation system is shown in Fig. 8. The *hypervisor* tracks individual VM state such as CPU utilization, disk I/O rate and so on. *Power meter* reads the aggregated power of the physical system. The are two steps for the power estimation: offline data collecting and online real-time estimation[4]. During the offline data collecting period, the partially measured state and power data are stored in a $v(\mathcal{S}, \mathcal{C})$ table. In the online real-time power estimation period, the *hypervisor* forwards VM coalition and states to the *VHC-based linear approximation* module and *VHC-based linear approximation* module looks up the corresponding $v(\mathcal{S}, \mathcal{C})$ in the table and estimates the unobserved $v(\mathcal{S}, \mathcal{C})$. Then $v(\mathcal{S}, \mathcal{C})$s are delivered to *Shapley value calculation* module to calculate individual VM power usage.

B. Prototype System

We have implemented the above framework and built a prototype system. The prototype consists of a power meter and two servers (A and B) with Intel Xeon 16-core CPU, 32GB memory and 2TB disks as shown in Fig. 9. The power meter uses a serial port to read the real-time electric signals including voltage, current, active power and so on. Server A runs the Citrix Xenserver 6.5 as the virtualized system and is connected to the socket of power meter. Server B reads the power information through the serial port with the sampling rate of $1Hz$. The power estimation part shown in the framework is also implemented on server B.

C. State Information Collection

We make use of the off-the-shelf `dstat` tool to collect the component states of a VM in real-time. In Section V-A, our solution considers different components (e.g., CPU, memory, disk etc). But according to our test, CPU power varies from 0 to 160 W, memory power is around 12 W and disk is around 10 W. Compared with the power of CPU, other components powers are stable during job running and negligible when distributed to each VM. Thus, we focus on the CPU utilization in the evaluation. But our method can be applied to different components.

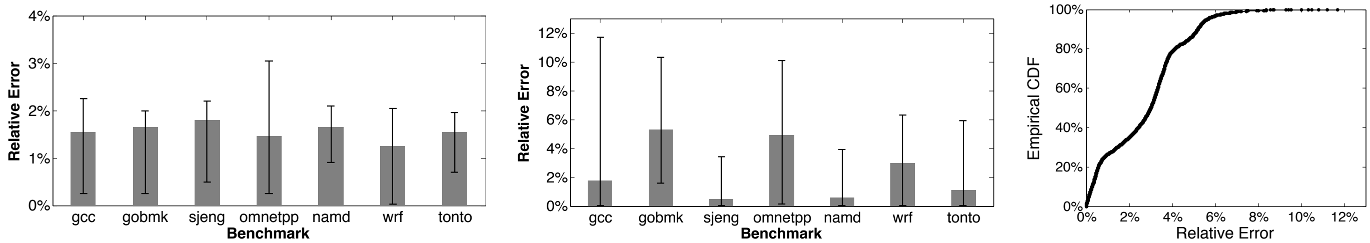
VII. Evaluation

Though Shapley value itself is provably fair [38], the unobserved $v(\mathcal{S}, \mathcal{C})$s, which are estimated by our VHC-based linear approximation, may not accurate, thus leading to the error of Shapley value. Hence, in this section, we first verify the accuracy of $v(\mathcal{S}, \mathcal{C})$ provided by our VHC-based linear approximation, and then make a detailed comparison for our Shapley value-based solution and other VM power estimation policies.

A. Experimental Setup

- Number of VHCs (r): According to the practical setup, we choose 4 types of VMs running on the prototype system, and the corresponding VM configurations are shown in Table IV. Therefore, we logically divide the VMs into 4 VHCs and build power model for each type of VM in isolation.

[4]The time interval between the two estimations depends on the sampling rate of VM states and power meter. In previous works, 1 second is treated as real-time estimation interval [20], [23], [24], which is also adopted in our evaluation.



(a) Errors of homogeneous coalition.

(b) Errors of heterogeneous coalition.

(c) Error distribution.

Fig. 10. Error analysis of the estimated $v(\mathcal{S}, \mathcal{C})$.

- Number of component states (c_i): As introduced in Sec. VI-C, we choose CPU utilization as the VM state.
- Normalizing resolution: We set the resolution of normalized CPU utilization data in each entry as 0.01.
- Sampling rate: The most fine-grained sampling rate for the power meter and operating system is $1Hz$, which has been widely applied for real-time sampling in previous works [20], [23], [24].
- Evaluation benchmark: As the CPU utilization is selected as the VM state, we utilize the a subset of *SPEC CPU 2006* suite as the validation benchmark which consists of integer and float point arithmetic [39], as shown in Table V. In addition, we also program a synthetic benchmark which randomly consumes CPU cycles to measure different $v(\mathcal{S}, \mathcal{C})$s.
- Idle power: Testing the prototype, we find that when all the VMs are shut down or idle, the physical machine's power is relatively stable, maintaining around $137 \sim 139$ W. Thus, in the following evaluations, we treat the physical machine's idle power as a constant (138 W) and deduct it in our power estimation.

TABLE IV
VM CONFIGURATION

VM Type	*vCPU*	*Memory*	*Disk*	*Power model*
VM_1	1	2G	20G	$p = 13.15u$
VM_2	2	4G	40G	$p = 22.53u$
VM_3	4	8G	80G	$p = 50.26u$
VM_4	8	14G	100G	$p = 96.99u$

B. Accuracy of $v(\mathcal{S}, \mathcal{C})$ Estimation

The estimated $v(\mathcal{S}, \mathcal{C})$ by our VHC-based linear approximation directly determines the accuracy of Shapley value. We next evaluate the accuracy of our VHC-based linear approximation.

We choose two coalitions as the representatives. One is composed of four VM_1-type VMs as the homogeneous coalition, and the other is composed of four different types of VMs as the heterogeneous coalition. By running our synthetic workload, we collect different $v(\mathcal{S}, \mathcal{C})$ of these two coalitions. According to the partially obtained $v(\mathcal{S}, \mathcal{C})$, we apply our VHC-based approximation to calculate the corresponding mapping vector $w_1 = 9.42$ for homogeneous coalition and $[w_1, w_2, w_3, w_4] = [16.98, 17.91, 23.42, 75.21]$ for heterogeneous coalition.

Then we run the SPEC CPU2006 benchmark shown in Table V for both coalitions and collect the VM states and physical machine's power. Using the above mapping vectors, we obtain the estimated $v(\mathcal{S}, \mathcal{C})$, where $\mathcal{C}$ corresponds to the VM states under these benchmarks. The relative errors of homogeneous case and heterogeneous case compared to the measured physical machine's power (deducting the idle power) are illustrated in Fig. 10(a) and Fig. 10(b), respectively.

To illustrate the performance results more clearly, we further analyze the relative errors in statistic and demonstrate the distribution in Fig. 10(c). As shown in the figures, the maximum relative error is 11.71%, but around 90% of the estimations are with the relative error less than 5%. For both cases, the estimations from our VHC-based approach in approximating the $v(\mathcal{S}, \mathcal{C})$ are all with average relative errors less than 5.33% under different benchmarks. Thus, our VHC-based approach is effective to approximate the Shapley value and guarantee the fairness.

C. Comparison of Different Power Estimation Policies

We next evaluate how well Shapley value performs compared to the other power estimation approaches. We initialize 5 VMs on the prototype system, including 2 VM_1-type VMs, 1 VM_2-type VM, 1 VM_3-type VM and 1 VM_4-type VM. First, we train the corresponding power models for each VM type shown in Table IV. As there are 4 types of VMs, thus we traverse all the 16 (2^4) combinations and derive

TABLE V
WORKLOADS/BENCHMARKS USED FOR SHAPLEY VALUE EVALUATIONS

Workload		*Description*	*Purpose*
SPECint	`gcc`	Compiler	Validation
	`gobmk`	Artificial Intelligence: go	
	`sjeng`	Artificial Intelligence: chess	
	`omnetpp`	Discrete Event Simulation	
SPECfp	`namd`	Biology/Molecular Dynamics	
	`wrf`	Weather Prediction	
	`tonto`	Quantum, Chemistry	
Our Synthetic		Occupy CPU randomly	Measure different $v(\mathcal{S}, \mathcal{C})$

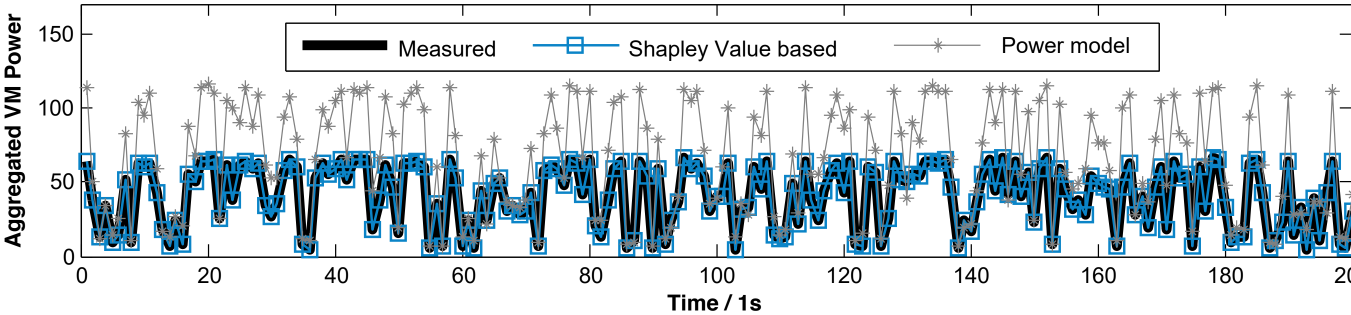


Fig. 11. Comparison of the performance in estimating the aggregated power (excluding the physical machines idle power).

the corresponding mapping vectors using our VHC-based approximation.

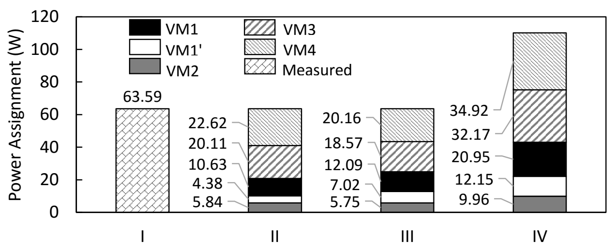


Fig. 12. Estimation results of the sample (I: Measured Aggregated Power, II: Shapley Value-based, III: Resource usage-based, IV: Power Model-based).

The real-time power of the physical machine (deducting the idle power) and the sum of VM power estimated by Shapley value and power model are all demonstrated in Fig. 11. We can clearly see that, the power model violates the macro-level accuracy and results in an average relative error of 56.43%. In contrast, the estimations by Shapley value-based solution are always consistent with the real measurements. Note that Shapley value always satisfies *efficiency* even the $v(\mathcal{S}, \mathcal{C})$s are not accurate.

As mentioned in Section IV-B, another naïve power estimation method, which also possesses the *efficiency*, is resource usage-based allocation (i.e., rescaling the aggregated power to individual VMs based on their resource usages). To look into the differences between Shapley value and the resource usage-based allocation, as well as the power model based method, we select a sample from Fig. 11 and illustrate the estimated results for each VM in Fig. 12. In fact, the resource usage-based allocation is an adjusted solution based on power model, so we can see that the proportions of power allocations are the same. Shapley value, however, gives different power allocations from them, which is fairer as interpreted in Section IV-B.

VIII. Open Challenges

Although our Shapley value based method is proved to be effective for VM power estimation, it is subject to some limitations, which pose open research challenges for our future investigation.

Micro-level accuracy: In the problem of VM power estimation, the micro-level accuracy cannot be validated when multiple VM co-exist in the same machine, due to the lack of ground truth of individual VM's power. Thus, the best we can do it to guarantee the macro-level accuracy and the fairness of power allocations to VMs. So far, It is still an open challenge to validate the micro-level accuracy.

Applicable scenario: To ease the resource management in large datacenters like Amazon EC2, VMs are categorized and each category is configured with fixed hardware resources. Our Shapley value-based method fits these datacenters well. In some other scenarios, however, VMs are configured with arbitrary hardware resources, leading to a large number of VM types. In this case, it might be difficult to apply our VHC-based linear approximation and new approximating approaches will be needed.

Accounting other power consumption: A running VM may depends on other resources, such as network devices and disk array, which do not belong to the local machine. Tracking the power of these non-local components and allocating their power to individual VMs pose another open challenge, and this would be our future research on extending the Shapley value based solution. We also would like to point out that there is no commonly-accepted method to account the idle power. Possible solutions include: (*i*) equally attributing the idle power to the running VMs, or (*ii*) proportionally attributing to VMs according to Φ_i. There are no convincing arguments that one way is better than the other. A practical solution may have to go through negotiation between the datacenter operator and end users.

IX. Conclusion

In this paper, we reveal the power interaction between VMs, which can lead errors of power model-based VM power estimation, and analyze how the modern computer system architecture causes such a problem. To overcome the drawbacks in existing solutions, we formulate the VM power accounting problem as a cooperative game. By proposing a non-deterministic Shapley value and developing a fast VHC-based linear approximation, we provide a fair VM power estimation. The experimental results on a prototype show that compared with the exact Shapley value, our non-deterministic Shapley value can achieve less than 5% error for 90% of the time, and it is fairer than resourced usage-based allocation.

REFERENCES

[1] NRDC, "Natural Resource Defense Council Report: America's Data Centers Are Wasting Huge Amounts of Energy," http://www.nrdc.org/energy/files/data-center-efficiency-assessment-IB.pdf.

[2] R. Nathuji and K. Schwan, "Vpm tokens: Virtual machine-aware power budgeting in datacenters," in *Proceedings of the 17th International Symposium on High Performance Distributed Computing (HPDC)*, ACM, 2008.

[3] Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in saas clouds," in *Proceedings of International Conference on Computer Communications (INFOCOM)*, IEEE, 2013.

[4] S. Govindan, D. Wang, A. Sivasubramaniam, and B. Urgaonkar, "Leveraging stored energy for handling power emergencies in aggressively provisioned datacenters," in *Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, ACM, 2012.

[5] D. Wang, C. Ren, and A. Sivasubramaniam, "Virtualizing power distribution in datacenters," in *Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA)*, ACM, 2013.

[6] D. Wang, S. Govindan, A. Sivasubramaniam, A. Kansal, J. Liu, and B. Khessib, "Underprovisioning backup power infrastructure for datacenters," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, ACM, 2014.

[7] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu, and H. Jin, "Carbon-aware online control of geo-distributed cloud services," *IEEE Transactions on Parallel and Distributed Systems*, 2016.

[8] Statista, "Average retail electricity prices in the US," http://www.statista.com/statistics/183700/us-average-retail-electricity-price-since-1990/.

[9] "Amazon EC2," https://aws.amazon.com/ec2/instance-types/.

[10] Statista, "Global electricity prices by select countries in 2015," http://www.statista.com/statistics/263492/electricity-prices-in-selected-countries/.

[11] "Amazon EC2 Instance Pricing," https://aws.amazon.com/ec2/pricing/.

[12] "Microsoft Azure Pricing Calculator," https://azure.microsoft.com/en-us/pricing/calculator/.

[13] M. A. Islam, H. Mahmud, S. Ren, and X. Wang, "Paying to save: Reducing cost of colocation data center via rewards," in *International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, 2015.

[14] "R. a. dines, "build or buy? the economics of data center facilities", in forrester research," http://www.io.com/wp-content/uploads/2013/04/build-or-buy.pdf.

[15] "Enaxis consulting, "pricing data center co-location services", 2009,," http://enaxisconsulting.com.

[16] C. Isci, S. McIntosh, J. Kephart, R. Das, J. Hanson, S. Piper, R. Wolford, T. Brey, R. Kantner, A. Ng, J. Norris, A. Traore, and M. Frissora, "Agile, efficient virtualization power management with low-latency server power states," in *Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA)*, ACM, 2013.

[17] Q. Wu, Q. Deng, L. Ganesh, C.-H. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, and Y. J. Song, "Dynamo: facebook's data center-wide power management system," in *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on*, 2016.

[18] D. Wong, "Peak efficiency aware scheduling for highly energy proportional servers," in *Proceedings of the 43rd International Symposium on Computer Architecture (ISCA)*, 2016.

[19] J. Stoess, C. Lang, and F. Bellosa, "Energy management for hypervisor-based virtual machines." in *Proceedings of the Conference on USENIX Annual Technical Conference (ATC)*, USENIX, 2007.

[20] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proceedings of the 1st ACM symposium on Cloud computing (SoCC)*, ACM, 2010.

[21] B. Krishnan, H. Amur, A. Gavrilovska, and K. Schwan, "Vm power metering: feasibility and challenges," *ACM SIGMETRICS Performance Evaluation Review*, 2011.

[22] A. E. H. Bohra and V. Chaudhary, "Vmeter: Power modelling for virtualized clouds," in *International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, IEEE, 2010.

[23] M. Colmant, M. Kurpicz, P. Felber, L. Huertas, R. Rouvoy, and A. Sobe, "Process-level power estimation in vm-based systems," in *European Conference on Computer Systems (EuroSys)*, ACM, 2015.

[24] G. Tang, W. Jiang, Z. Xu, F. Liu, and K. Wu, "Zero-cost, fine-grained power monitoring of datacenters using non-intrusive power disaggregation," in *Proceedings of the 9th International Conference on Middleware*, ACM/IFIP/USENIX, 2015.

[25] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of Annual International Symposium on Computer Architecture (ISCA)*, ACM, 2007.

[26] W. L. Bircher and L. K. John, "Complete system power estimation using processor performance events," *Transactions on Computers (TC)*, IEEE, 2012.

[27] D. A. Ellsworth, A. D. Malony, B. Rountree, and M. Schulz, "Dynamic power sharing for higher job throughput," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, ACM, 2015.

[28] Y. Zhai, X. Zhang, S. Eranian, L. Tang, and J. Mars, "Happy: Hyperthread-aware power profiling dynamically," in *Proceedings of the Conference on USENIX Annual Technical ConferenceUSENIX (ATC)*, USENIX, 2014.

[29] W. Huang, C. Lefurgy, W. Kuk, A. Buyuktosunoglu, M. Floyd, K. Rajamani, M. Allen-Ware, and B. Brock, "Accurate fine-grained processor power proxies," in *Proceedings of the 45th Annual International Symposium on Microarchitecture*, IEEE/ACM, 2012.

[30] R. Koller, A. Verma, and A. Neogi, "Wattapp: an application aware power meter for shared data centers," in *Proceedings of the 7th international conference on Autonomic computing (ICAC)*, ACM, 2010.

[31] V. Misra, S. Ioannidis, A. Chaintreau, and L. Massoulié, "Incentivizing peer-assisted services: a fluid shapley value approach," in *ACM SIGMETRICS Performance Evaluation Review*, 2010.

[32] R. Stanojevic, N. Laoutaris, and P. Rodriguez, "On economic heavy hitters: shapley value analysis of 95th-percentile pricing," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC)*, ACM, 2010.

[33] M. Dong, T. Lan, and L. Zhong, "Rethink energy accounting with cooperative game theory," in *Proceedings of the 20th annual international conference on Mobile computing and networking (MOBICOM)*, ACM, 2014.

[34] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron, "Powerpack: Energy profiling and analysis of high-performance systems and applications," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2010.

[35] A. Kansal and F. Zhao, "Fine-grained energy profiling for power-aware application design," *ACM SIGMETRICS Perform. Eval. Rev.*, 2008.

[36] H. Lim, A. Kansal, and J. Liu, "Power budgeting for virtualized data centers," in *Proceedings of the Conference on USENIX Annual Technical Conference (ATC)*, USENIX, 2011.

[37] R. Shea, H. Wang, and J. Liu, "Power consumption of virtual machines with network transactions: Measurement and improvements," in *International Conference on Computer Communications (INFOCOM)*, IEEE, 2014.

[38] L. S. Shapley, "A value for n-person games," Tech. Rep., 1952.

[39] A. Phansalkar, A. Joshi, and L. K. John, "Subsetting the spec cpu2006 benchmark suite," *SIGARCH Comput. Archit. News*, 2007.