



A frequency-aware management strategy for virtual machines in DVFS-enabled clouds

Jianzhou Mao ^{*}, Xiaopu Peng, Ting Cao, Tathagata Bhattacharya, Xiao Qin

Department of Computer Science and Software Engineering, Auburn University, Auburn 36830, AL, USA

ARTICLE INFO

Keywords:

Cloud computing
DVFS
Energy optimization
Static power
Dynamic power
Virtual machines
Frequency ratio

ABSTRACT

It is demanding to curtail energy consumption of virtual-machine-powered data centers, because modern data centers have been significantly scaling up in capacity in past decades. In this study, we propose a frequency-aware management strategy, which controls dynamic power and static power of processors running virtual machines in data centers. Unlike existing dynamic voltage and frequency scaling schemes, our strategy simply incorporates frequency requirements rather than task execution times. This salient feature is practical, because task execution times in a raft of real-world applications are unknown in a priori. We build a frequency-aware model, which is adept at deriving an optimal frequency ratio that minimizes processors' energy consumption. With our model in place, the energy efficiency of a data center can be maximized by adjusting the processor's frequency to meet the optimal frequency ratio. We design a management approach to judiciously adjust frequency ratio to conserve energy without violating the frequency requirements imposed by virtual machines. After analyzing the correlations between frequency ratio and energy consumption, we show that a small static power proportion gives rise to high energy-saving performance. The results demonstrate that our model lays out a solid theoretical foundation catering to the development of power management software in DVFS-enabled clouds.

1. Introduction

The performance of cloud computing platforms has been significantly growing in the past decade. With the dramatic evolution in computing capacity, the power consumption of datacenters customized for cloud computing is skyrocketing. In this study, we propose to seamlessly integrate DVFS (i.e., Dynamic Voltage and Frequency Scaling) scheduling with virtual machine management to effectively conserve energy consumption in datacenters.

Our frequency-aware scheduler aims to minimize CPU energy cost in virtual-machine-enabled datacenters by configuring the most appropriate CPU frequency ratios according to workload conditions. Four emerging trends below strongly motivate us to contrive this frequency-aware DVFS scheduling algorithm.

- Virtual-machine-based cloud computing platforms are technical underpinnings for modern datacenters in the future.
- Large-scale datacenters have a pressing demand to be economically and environmentally friendly.

- The dynamic voltage and frequency scaling technique or DVFS is one of the most practical energy-saving technologies.
- Quality of service (i.e., QoS) opens a door for optimizing energy efficiency without violating service-level agreements.

In our expanding digital world, data are changing the way we live, work, and entertain. International Data Corporation or IDC speculates that the aggregated data in the world will grow from 33 ZB in 2018 to 175 ZB by 2025 at a significant annual growth rate of 61% [1]. To meet the accommodate such a massive amount of data, the scale of datacenters is snowballing to reach an unbelievable level. The global data-center market is estimated to exceed \$174 billion by 2023, growing at an annual rate of approximately 4% during the forecast period. The largest companies such as Facebook, Google, Amazon, and Microsoft are focusing on the development of modular and hyperscale data-center construction facilities [2].

Cloud-based datacenters have become a new trend of the enterprise data repository replacing traditional datacenters. Cloud computing offers promising benefits such as enhanced scalability, efficiency,

^{*} Corresponding author.

E-mail address: jzm0122@auburn.edu (J. Mao).

<https://doi.org/10.1016/j.suscom.2021.100643>

Received 15 April 2021; Received in revised form 27 August 2021; Accepted 25 November 2021

Available online 3 December 2021

2210-5379/© 2021 Elsevier Inc. All rights reserved.

flexibility of business operations, and to name just a few. A growing number of large companies have chosen to utilize cloud datacenters (e.g., Microsoft Clouds and Amazon Web Services) to store enormous amounts of data. IDC predicts that 49% of the world's stored data will reside in public cloud environments in year 2025 [11]. A report published by Cisco forecasts that 94% of all workloads and computing will be placed in cloud datacenters in 2021 [3]. More specifically, overall workloads and compute instances in datacenters will be doubled (i.e., 2.3-fold) from 2016 to 2021. When it comes to computing clouds, the workloads are expected to expand by a factor of 2.7-fold during the same period.

Virtual machines are a driving force behind the wide adoption of cloud-based datacenters. Virtual machines – software-based machine emulations – facilitate flexible, cost-effective, and on-demand computing environments. Virtual machines make it feasible to save a lot of energy by increasing workloads and compute instance density. The workload and compute instance density of cloud servers is expected to grow from 8.8 in 2016 to 13.2 by 2021. In comparison, the workload and compute instance density in traditional data center servers will merely increase from 2.4 in 2016 to 3.8 in the same period [3]. Moreover, virtual machines are positioned to support legacy software and computing environments [4].

The energy consumption of these large-scale datacenters is truly tremendous. A long-term goal of this study is to devise resource management tools for future economically and environmentally friendly datacenters. Servers, core IT components in datacenters, are major players contributing to high energy cost (e.g., more than 26%) [5]. CPUs, main memory, and disk I/Os are key contributors to energy cost in servers. The energy cost of CPU, in turn, occupies the highest proportion in the server (i.e., 42%) [6]; in some scenarios, CPUs account for up to 70% of total energy consumption in servers [7]. Thus, conserving CPU energy is indispensable in tackling the datacenter energy problem. In a handful of prior studies [8], CPU energy saving policies have been investigated without considering main memory or I/Os and; therefore, we followed this research methodology to place the CPU energy efficiency under the microscope.

VM consolidation and DVFS (Dynamic Voltage and Frequency Scaling) are two popular energy conservation methods adopted in data centers. The VM consolidation technique reduces the number of physical servers by migrating VMs and shutting down idle servers to save energy consumption [9,10]. Unlike VM consolidation, DVFS is beneficial when servers exhibit low idle power consumption, high utilization, and large VM migration overhead. DVFS enables processors to consume less power by adjusting to the most appropriate frequency and supplied voltage. Almost all processors are built in CMOS circuit and; therefore, evidence shows that energy consumption of CPU is approximately proportional to frequency and the square of the voltage. Decreasing voltage and processor frequency will, of course, scale down computing performance and stretch executing times. This performance problem is well addressed by incorporating the concept of quality of service (a.k.a., QoS) while adjusting voltage and CPU frequency. The popularity of the DVFS technique motivates us to explore an optimization strategy to enhance the energy efficiency of DVFS-enabled datacenters.

When deploying the DVFS technique, one has to be aware of static power consumption that is incurred by leakage current in CMOS devices. More often than not, DVFS inevitably extends execution times at the cost of increased static power consumption. We advocate a holistic solution that orchestrates both dynamic power and static power management. More specifically, we propose a model that takes into account static and dynamic power consumption. With the proposed model in place, we conduct an experiment to evaluate the energy efficiency of servers running multiple virtual machines in datacenters.

To build energy-efficient datacenters, it is prudent to make a good tradeoff between energy efficiency and performance. A practical energy-saving method is to conserve energy while meeting Service Level Agreements (a.k.a., SLA requirements) [11]. Improving the energy

efficiency of datacenters maximizes profits by significantly cutting back energy consumption. It is conventional wisdom to incorporate execution time and deadlines into QoS requirements of submitted tasks. In a raft of real-world scenarios (e.g., web crawlers), task execution times are unknown a priori. This constraint motivates us to pilot a novel model that does not rely on task execution times and deadlines.

In our design, we take into account frequency requirements as part of SLAs. In cloud computing platforms like Google Cloud and Microsoft Azure, users submit the required number of virtual-machine instances accompanied by frequency requirements. Since cloud-computing platform cost is proportional to the number of VMs as well as required frequency, users strive to make the most appropriate tradeoff between performance and cost. We propose a new QoS model, where minimum frequency requirements is a core component. In doing so, users have no obligation to specify execution times and deadlines in QoS; rather, frequency requirements can straightforwardly serve the purpose. Our energy conservation scheme fosters energy efficiency while striving to meet frequency requirements rather than deadlines. QoS requirements will be guaranteed if configured frequencies are the same or higher than the specified minimum frequencies.

Compared with existing solutions, our approach has the following three salient strengths. First of all, frequency ratios are used as a metric to specify QoS requirements and energy consumption rather than the execution time parameter. Our model derives optimal energy savings for servers without estimating execution times or specifying deadlines. Second, our model offers optimal frequency ratios customized for various processors in accordance to their static power proportions, thereby maximizing the processors' energy efficiency. Last, our frequency-aware DVFS model, being practical in nature, is readily to be adopted by power management systems. Given a server's static power and max dynamic power, the model can govern a power manager to achieve high energy efficiency.

In a nutshell, the main contributions of this paper are summarized as follows:

- We design a QoS-base strategy that adjusts frequency ratios to optimize energy efficiency.
- We build a model that offer optimal frequency ratios customized for various processors in accordance to static power proportions.
- We investigate the intriguing features of the frequency-ratio-based DVFS model.
- We apply the novel model to measure the energy efficiency of virtual-machine-powered servers in data centers.

The remainder of this paper is organized as follows. Section 2 outlines the previous studies related to power modeling, energy conservation techniques, DVFS, and task models. Section 3 describes a new frequency-aware QoS model, in which QoS requirements are represented using CPU frequencies rather than deadlines. The concept of frequency ratio accompanied by the frequency-aware DVFS model are proposed in Sections 4 and 5, respectively. The analysis of the DVFS model with respect to frequency ratio can be found in Section 6. Section 7 discusses the sample usages and applicability of the proposed frequency-aware model. We conclude this paper with future work in Section 8.

2. Related work

This section outlines the related work of the energy conservation techniques and QoS models of cloud datacenters.

2.1. Power modeling of cloud datacenters

Prior studies have been focused on energy-consumption monitoring and management of datacenters from a global perspective [12]. In order to monitor and manage cloud servers in a much more meticulous view,

researchers built up models to gauge the energy consumption of hosts. For example, Roy et al. [13] demonstrated that server energy consumption is the summation of CPU and RAM energy consumption. Tudor et al. [14] incorporated I/O energy consumption into an energy-consumption model.

In addition to the above cumulative models, a handful of practical models shed light on the relationship between CPU utilization and total energy consumption of servers. For instance, Inozahy et al. [15] discovered the energy consumption models of servers under various CPU frequencies. Following-up studies [16,17] investigated that energy consumption is linearly proportional to CPU frequency. Such linear-like models are popular and practical thanks to the advantages of simplicity and high accuracy. Therefore, in this study we adopt the linear-like modeling method to build a new model for DVFS-enabled clouds.

2.2. Energy conservation techniques

In the past decade, a growing number of energy-saving methods have been devised to conserve the energy consumption in datacenters [18, 19]. For example, Hao et al. [20] proposed a real-time virtual machine placement algorithm customized for distributed datacenters. This placement algorithm maximizes service capabilities by adjusting virtual machine deployment, thereby enabling datacenters to serve more users with short response times. Nguyen and Chefriet [21] designed an algorithm to optimize energy consumption by taking into account both the requirements of computing capacity and bandwidth of virtual machines, as well as the capacity constraints of physical computing resources and bandwidth resources. Mei et al. [8] developed a virtual machine placement strategy in a hybrid cloud environment. The goal of this strategy is to maximize the profit of service provider through two adjustments, namely, (1) changing the deployment locations of virtual machines and (2) configuring the numbers of local servers and rental servers. Different from the aforementioned schemes that belong to non-DVFS techniques, our proposed strategy is focused on profiling tools for virtual machines running in DVFS-enabled data centers.

2.3. Dynamic voltage and frequency scaling

Dynamic voltage and frequency scaling or DVFS is a widely adopted energy-conservation technique for clusters housed in data centers. Much attention has been paid toward DVFS-based task schedulers that minimize the total energy consumption without violating deadlines [22]. Aydin and Yang [23] showed that task scheduling on multiprocessors to minimize energy consumption is an NP-Hard problem in the strong sense. As such, a heuristic EDF (i.e., Earliest Deadline First) scheduler was devised to minimize CPU energy of multiprocessors. Wu et al. [24] applied DVFS to reduce the energy consumption of a server during the course of idle or light workload. Unlike the above DVFS techniques that are focused on dynamic CPU power, our model pay heed on CPU static power consumption due to CMOS circuits.

2.4. Task models for real-time computing

When it comes to real-time tasks, corresponding QoS requirements ought to be satisfied when conserving system energy consumption. Task models become an underpinning component of energy-efficient schedulers for real-time tasks [25]. For instance, Wang et al. [26] constructed a model, in which tasks have various priorities coupled with task-precedence constraints. In some cases, a group of tasks are represented in the form of a weighted directed acyclic graph or DAG. In another model, an array of tasks share a common deadline [27,28]; these tasks have diversified worst case execution time or WCET. Aydin and Yang built a periodic task model, for which a scheduling algorithm improves energy efficiency [23]. With the tasks models in place, numerous scheduling algorithms were implemented for saving energy of computing systems running real-time applications [29–31]. In contrast

to the existing task models, ours incorporates frequency requirements rather than relying on task execution times. Our model addresses a critical issue in a raft of real-time applications, where task execution times are unknown in a priori.

3. Frequency-aware QoS

Recall that the DVFS conserves energy consumption by lowering processor frequency. Scaling down processor frequency inevitably increases task execution times, which may incur violations in service level agreements or SLA. In this section, we shed some light on the concept of worst case execution time in Section 3.1. Next, we discuss a way of applying our model to satisfy QoS requirements in Section 3.2.

3.1. Worst case execution time

From the perspective of QoS modeling, virtual machines are treated as standalone tasks running on servers. As such, we use terms virtual machines and tasks interchangeably throughout this section. Table 1 lists the notation used throughout this paper.

In real-time task modeling, it is a conventional wisdom to adopt WCET (i.e., Worst case execution-time) and deadlines (i.e., upper timing bound) to characterize QoS requirements of real-time applications [32]. Fig. 1 shows a traditional QoS model that shed light on the relationships among CPU frequency, WCET time, and deadlines. Task τ_1 and τ_2 , sharing the same deadline, have different WCET requirements measured in clock cycles. These two tasks have different execution time under the same CPU frequency. If no DVFS energy-saving technology is deployed, server will run at the its full capacity using the max frequency level f^{max} . In this case, τ_1 and τ_2 finish at *FinTime1* and *FinTime2*, which must be earlier than the deadline. The last sub-figure in the bottom of Fig. 1 depicts that DVFS scales the CPU frequency from f^{max} down to f_a . If τ_1 and τ_2 both run on this server concurrently, each task's execution time is extend. Nevertheless, it is mandatory to make finish time *FinTimeNew* earlier than the deadline.

3.2. Frequency-aware QoS modeling

Apart from WCET and deadlines, CPU frequency is an exceptional and practical parameter to capture the QoS requirement of real-time tasks. Giving memory and I/O resources, task execution times largely depends on processors and the elected CPU frequency level. Therefore, it is arguably feasible to transfer a time-aware requirement into a frequency-aware requirement.

Fig. 2 depicts two types of real-time tasks coexisting in our frequency-aware QoS modeling. For traditional real-time tasks (see the tasks on the top side of Fig. 2), real-time tasks should entail timing

Table 1
Symbol and annotation.

Symbol	Annotation
f_c^{opt}	Optimal energy-saving frequency for processor c applied our model.
$f_{i,c}^{req}$	Frequency requirement of virtual machine vm_i on processor c
$f_{V,c}^{req}$	Frequency requirement of virtual machine set V on processor c .
	V is the set of virtual machine running on processor c .
$f_{V,c}^{conf}$	Frequency configured for virtual machine set V on processor c
Γ_i	Total execution requirements of virtual machine vm_i .
$\gamma_{i,j}$	Execution requirements of the j th segment of virtual machine vm_i .
p_c^{sta}	Static power of processor c .
p_c^{dmax}	Max dynamic power of processor c .
r_c	Frequency ratio of processor c .
f_c^{max}	Max frequency level of processor c .
E_c^{opt}	The optimal energy consumption of processor c .
E_C^{opt}	The optimal energy consumption of processor set C .

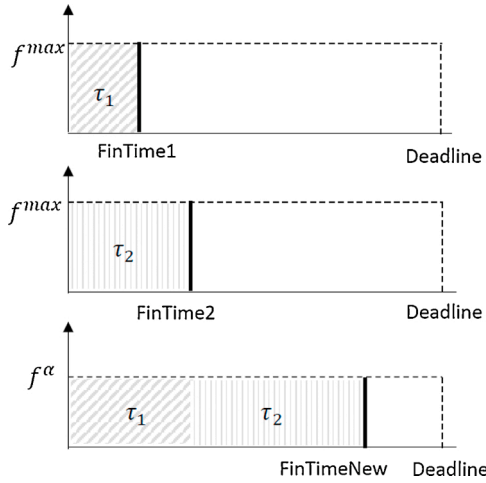


Fig. 1. Traditional time-aware QoS model.

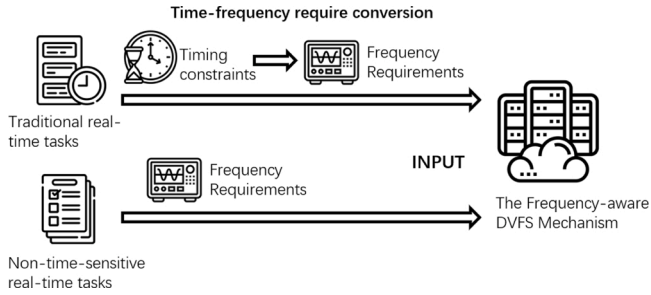


Fig. 2. Frequency-aware modeling for time-sensitive and non-time-sensitive real-time tasks running in DVFS enabled systems.

constraints like deadlines and WCET values, and we transfer the time-aware requirements to the frequency-aware requirements (e.g., minimum frequency requirements). When it comes to non-time-sensitive tasks like long-running applications (see the tasks on the bottom side of Fig. 2), non-time-sensitive applications merely pass on frequency requirements to the DVFS-enabled system.

In our frequency-aware QoS model, we propose to maintain a processor frequency higher than the summation of the minimum frequency requirements of all the tasks running on the processor. Fig. 3 shows our frequency-aware QoS model for traditional tasks, in which task τ_1 and τ_2 have the same settings as those in Fig. 1. Frequency $f_1^{\min}$ and $f_2^{\min}$ are the

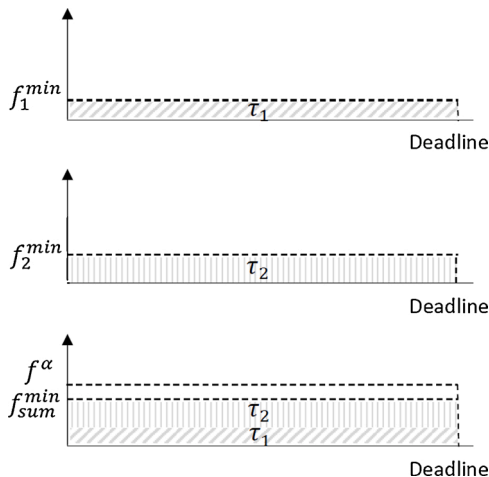


Fig. 3. Our frequency-aware QoS model for traditional real-time tasks.

minimum frequency that averts SLA violations in τ_1 and τ_2 governed by DVFS. As the execution time of τ_1 and τ_2 extend to the max time task allowed, $f_1^{\min}$ and $f_2^{\min}$ are no doubt lower than $f^{\max}$. When τ_1 and τ_2 both run on this processor, $f_{\text{sum}}^{\min}$ is the new minimum frequency requirement that is higher than $f_1^{\min}$ and $f_2^{\min}$. In this scenario, the most appropriate energy-saving frequency f^a is higher than $f_{\text{sum}}^{\min}$. As such, we choose f_a as the processor frequency; Otherwise, we should use the sum of the minimum frequency requirements as the processor frequency. In a nutshell, this model allows servers to undertake real-time tasks using frequency requirements rather than traditional WCET and deadlines.

In real-world scenarios, there might be the lack of specified requirements in terms of deadline or WCET in submitted tasks. This phenomenon is specially true when it comes to long-running tasks like crawlers and web services. For such non-time-sensitive tasks, a system can render QoS by satisfying their minimum frequency requirements rather than WCET or deadlines (see Fig. 2). After taking into account minimum frequency requirements, our model facilitates the comparison between the sum of tasks' minimum frequency requirements running on the same processor and the best energy-saving frequency of that CPU. Then, we specify a higher one to avoid potential SLA violations.

We consider multiple independent aperiodic virtual machines running on a group of processors $C = \{c_1, c_2, \dots, c_m\}$. For simplicity without the loss of generality, we investigate a set of n independent aperiodic virtual machines $V = \{vm_1, vm_2, \dots, vm_n\}$ running on processor c (i.e., $c \in C$). Each virtual machine is represented by a pair, namely, $vm_i = (a_i, f_i^{\text{req}})$, where a_i is the arrival time of virtual machine vm_i , f_i^{req} is the minimum frequency requirement of virtual machine vm_i . As $f_{\text{sum}}^{\min}$ is the summation of frequency requirements $f_1^{\min}$ and $f_2^{\min}$ in Fig. 3, the relation between overall frequency requirement and each virtual machine frequency requirement is:

$$f_{V,c}^{\text{req}} = \sum_{i \in V} f_{i,c}^{\text{req}}. \quad (1)$$

If $f_{V,c}^{\text{conf}} \geq f_{V,c}^{\text{req}}$ holds, then processor c is able to perform tasks without SLA violations. Therefore, after we obtain the summation of minimum frequency requirement on processor c , we compare the summation with the energy-saving frequency f_c^{opt} derived from our DVFS algorithm (see Section 5). If the overall frequency requirement $f_{V,c}^{\text{req}}$ is lower than f_c^{opt} , we should elect the derived optimal energy-saving frequency. If $f_{V,c}^{\text{req}}$ is higher or equal to f_c^{opt} , we should guarantee the overall minimum frequency requirement in order to satisfy the QoS requirement of c performing virtual machine set V .

$$f_{V,c}^{\text{conf}} = \max(f_c^{\text{opt}}, f_{V,c}^{\text{req}}). \quad (2)$$

4. Frequency ratio modeling

Nowadays, modern processors are powered by the CMOS technology. The energy consumption of a CMOS circuit is composed of dynamic energy consumption E^{dyn} and static energy consumption E^{sta} [33]:

$$E = E^{\text{sta}} + E^{\text{dyn}}. \quad (3)$$

Prior studies [27,34] confirmed that dynamic power P^{dyn} is computed as

$$P^{\text{dyn}} = C_l \cdot A \cdot V_d^2 \cdot f, \quad (4)$$

where C_l is load capacitance, A is the percentage of active gates, V_d is supply voltage, and f is processor frequency.

The voltages of CMOS processors affect processor frequencies. For example, decreasing voltages would enlarge circuit delay, changing the clock frequency [35]. The correlation between frequency and voltage is expressed as:

$$f = k \cdot \frac{(V_{dd} - V_{th})^2}{V_{dd}}, \quad (5)$$

where V_{th} – typically a small value – is a threshold voltage [36]. Approximately, there is a linear relation between frequency and voltage (i.e., $f \approx k \cdot V_{dd}$). Because C_l , A , k are constants, we consolidate these three constants as $\beta = (C_l \cdot A)/k$. Thus, the CPU dynamic power consumption is rewritten as

$$P^{\text{dyn}} = \beta \cdot f^3. \quad (6)$$

We use max dynamic power P^{dmax} to represent the dynamic power when the processor frequency is set to a maximum level f^{max} . Thus, we derive (7) from (6) as

$$P^{\text{dyn}} = P^{\text{dmax}} \cdot \frac{f^3}{(f^{\text{max}})^3}. \quad (7)$$

Combined with CPU static power P^{sta} – a constant irrelevant to the voltage and frequency, the total CPU power is represented in the format of frequency as

$$P = P^{\text{sta}} + P^{\text{dmax}} \cdot \frac{f^3}{(f^{\text{max}})^3}. \quad (8)$$

It should be noted that when using DVFS, not all frequencies ranging from 0 to f^{max} are available due to discrete frequency levels like 10 % f^{max} , 20 % f^{max} , 40 % f^{max} , 100 % f^{max} [37]. We introduce *frequency ratio* r as a ratio between the current processor frequency f and the maximum frequency f^{max} held by a processor. Thus, we have $r = f/f^{\text{max}}$. The total CPU power can be obtained from frequency ratio r as

$$P = P^{\text{sta}} + P^{\text{dmax}} \cdot r^3. \quad (9)$$

5. Frequency-aware DVFS

Now we are in a position to present a DVFS model tailored for tasks issuing frequency requirements (see Fig. 2). After proposing the DVFS model in Section 5.1, we shed bright light on the overhead of this model in Section 5.2.

5.1. Modeling frequency-aware DVFS

If virtual machine vm_i is executed at frequency f_i in time t_i , Γ_i is the total number of clock cycles of vm_i . The energy consumption of virtual machine vm_i is derived from clock cycles Γ_i and frequency f_i as

$$E_i = P_i \cdot t_i = P_i \cdot \frac{\Gamma_i}{f_i}. \quad (10)$$

Power consumption is an instantaneous value depending on frequency in per time unit. We assume that virtual machine vm_i 's execution consists of K_i segments, where the clock cycles of the j th segment ($1 \leq j \leq K_i$) is γ_{ij} . Thus, the clock cycles of all the segments of virtual machine vm_i form a frequency set $\{\gamma_{i,1}, \gamma_{i,2}, \dots, \gamma_{i,K_i}\}$, which determines the total execution requirements Γ_i of virtual machine vm_i . Thus, we have

$$\Gamma_i = \sum_{j=1}^{K_i} \gamma_{ij}. \quad (11)$$

Let $f_{i,j}$ be the execution frequency for the j th execution segment of virtual machine vm_i . Energy consumption of the virtual machine can be obtained from Eq. 10 orchestrating the power, clock cycles, and frequency as

$$E_i = \sum_{j=1}^k P_{i,j} \frac{\gamma_{ij}}{f_{i,j}}. \quad (12)$$

The *Lagrange* multiplier method is applied to minimize energy E_i [38]. In case where frequencies of vm_i during all its intervals are

identical, the computation of (12) will be simplified. Furthermore, the common execution frequency f_i is equal to the overall execution requirement of vm_i divided by the overall execution time of vm_i . Thus, we have

$$f_i = f_{i,1} = f_{i,2} = f_{i,3} = \dots = f_{i,K}. \quad (13)$$

Hence, the optimal frequency of each segment is equal to the optimal solution for the entire virtual machine vm_i . Combining with Eqs. (9) and (10), if there are T VMs running on the processor c , we derive energy consumption of processor c as

$$E_c = \sum_{i \in T} \Gamma_i (P_c^{\text{sta}} + P_c^{\text{dmax}} (r_c)^3). \quad (14)$$

$$E_c = \sum_{i \in T} \frac{\Gamma_i}{f_c^{\text{max}}} \left(\frac{P_c^{\text{sta}}}{r_c} + P_c^{\text{dmax}} (r_c)^2 \right).$$

(14) is a quadratic function with respect to frequency ratio r_c . To obtain the extreme value of the energy consumption, we take the derivative of E_c with respect to r_c

$$E'_c = \sum_{i \in T} \frac{\Gamma_i}{f_c^{\text{max}}} \cdot \frac{1}{(r_c)^2} (2 \cdot (r_c)^3 \cdot P_c^{\text{dmax}} - P_c^{\text{sta}}). \quad (15)$$

To investigate the trend of this quadratic function parabola, let us take the second derivative of E_c with respect to r_c

$$E''_c = \frac{2 \cdot \sum_{i \in T} \Gamma_i}{f_c^{\text{max}}} \cdot \left(\frac{P_c^{\text{sta}}}{(r_c)^3} + P_c^{\text{dmax}} \right). \quad (16)$$

Since we have $E''_c > 0$, it is verified that E_c is a convex function of r_c and the parabola opens upward. From the features of parabola, by setting the first derivation of E'_c to zero, we can obtain the most energy-efficient consumption E_c^{opt} at the optimal frequency ratio f_c^{opt} . The optimal frequency ratio r_c^{opt} is derived from P_c^{sta} and P_c^{dmax} as

$$r_c^{\text{opt}} = \sqrt[3]{\frac{P_c^{\text{sta}}}{2P_c^{\text{dmax}}}}. \quad (17)$$

Noticing that f is discrete, we choose the nearest r value to meet deadline requirements. The optimal energy consumption of processor c is E_c^{opt} , which can be calculated as follows.

$$E_c^{\text{opt}} = \sum_{i \in T} \frac{\Gamma_i}{f_c^{\text{max}}} \left(\frac{P_c^{\text{sta}}}{r_c^{\text{opt}}} + P_c^{\text{dmax}} \cdot (r_c^{\text{opt}})^2 \right). \quad (18)$$

After optimizing energy consumption E_c^{opt} for a single processor (i.e., c), we derive optimal energy consumption E_c^{opt} for computing systems equipped with multiple processors (i.e., $C = \{c_1, \dots, c_m\}$) in a datacenter. More specifically, energy E_c^{opt} is an accumulative value of the optimal energy consumption of all the processors in set C . Thus, the system level energy consumption can be expressed as:

$$E_C^{\text{opt}} = \sum_{c \in C} E_c^{\text{opt}}. \quad (19)$$

where E_c^{opt} on the right-hand side of the expression is obtained from (18).

It is noteworthy that in our frequency-aware DVFS model, virtual machine energy consumption is independent of virtual machine execution time. Rather, the energy consumption is reliance to virtual machines' execution requirements and server hardware parameters (e.g., static power and max dynamic power.)

When the current frequency ratio is configured to r^{opt} , the energy consumption will be minimized to E^{opt} . There is a challenging roadblock to setup r to value r^{opt} due to the discrete value of processor frequency in the real world. If the current frequency ratio is less than r^{opt} , the energy consumption will be proportional to the frequency ratio. On the contrary, if the current frequency ratio is greater than r^{opt} , the energy

consumption will be inversely proportional to the frequency ratio.

The optimal execution time of processor c is t_c^{opt} governed by the optimal frequency ratio r_c^{opt} . Hence, we express time t_c^{opt} as a function of frequency ratio r_c^{opt} in (20).

$$t_c^{\text{opt}} = \frac{\sum_{i \in T} \Gamma_i}{f_c^{\text{max}}, r_c^{\text{opt}}} \quad (20)$$

5.2. Overhead analysis

An optimal frequency ratio is obtained from our proposed mathematical model, which swiftly yields a result thanks to its low cost. We analyze the overhead of the model from two perspectives, namely, time complexity and time/energy overhead.

- **Time complexity.** Given task set T that consists of a total number n of tasks, the time complexity of the computing process to execute our model (see Eq. 17) is $O(n)$, where n is the number of tasks.
- **Computing time and energy cost.** We conduct a case study to gauge the computing time of searching optimal frequency ratios for a set of 10,000 tasks. In this case study, we run the model on a Linux machine having a quad-core Intel i7-7820H 3.90 GHz, and 8 GB RAM. We run the overhead test 10 times. After omitting the maximum and minimum measures, we report the average results. The computing time and energy overhead for our model to output optimal frequency ratios are 13.3 ms and 0.585 Joule, respectively.

In a nutshell, the model articulated in Section 5.1 exhibits fairly low overhead in terms of computing time and energy consumption. The overhead analysis asserts that the cost spent in searching optimal frequency ratios can be ignored when the model is deployed in energy management modules for DVFS-enabled clouds.

6. Analysis of frequency ratios

Recall that in our proposed frequency-aware DVFS model (see Section 5), optimal energy consumption derived from (18) is dependent of frequency ratio r . In this section, we place the spotlight on the trend and features of the optimization formula by applying the DVFS model to various theoretical cases. Such an analysis validates the versatility of our new model with frequency awareness.

6.1. Energy consumption and frequency ratio

Eq. (14) indicates the correlation between energy consumption and

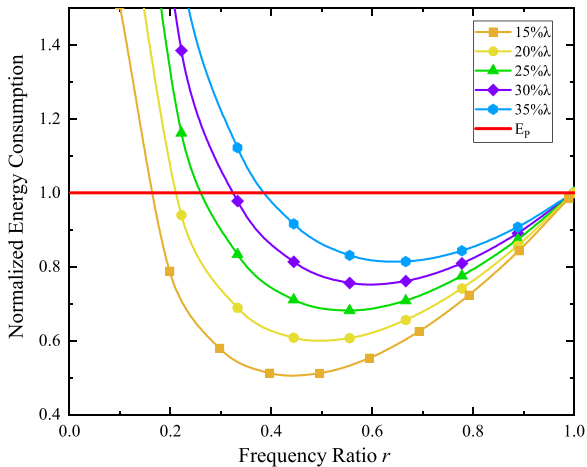


Fig. 4. Impacts of frequency ratio on normalized energy consumption under various λ value.

frequency ratio r . Fig. 4 depicts the normalized energy consumption E as a function of frequency ratio r of processors. This analysis is focused on a single processor. Nevertheless, the findings can be readily applied to scenarios where a computer system is orchestrating multiple processors.

In order to make the trends plotted in Fig. 4 broadly representative, we introduce a parameter called *static power proportion* in lieu of a server's power data. Eq. (14) suggests that the most vital parameters in the DVFS model is server static power P^{sta} and max dynamic power P^{dmax} – two constants. We simplify these two parameters by defining static power proportion λ to gauge static power P^{sta} as a percentage of the total power consumption in max frequency level. Thus, we have $\lambda = P^{\text{sta}} / (P^{\text{dmax}} + P^{\text{sta}})$. More often than not, a server's static power proportion λ is anywhere between 15% and 35% [39,40]. We categorize all types servers in terms of λ values, thereby allowing us to capture the energy consumption features of a wide range of general cases.

Fig. 4 shows five parabolas of the function expressed in Eq. (14) in a variety of cases. The red line represents a server's energy consumption when the server keeps running on at the peak power with the maximal frequency level all the time (i.e., frequency ratio $r = 1$). We refer to the energy consumption plotted by the red line as peak energy consumption or E_p ; in this scenario, no DVFS is involved in conserving energy. For comparison purpose, we normalize all the energy consumption derived from the model, within which E_p turns to be 1 – an ideal baseline. The five curves in Fig. 4 are the normalized energy consumption when the static power proportion λ is configured to 15%, 20%, 25%, 30%, and 35%, respectively. Fig. 4 depicts the energy consumption outcomes of the five λ values that cover a majority of real-world server cases.

These five parabolas share a similar pattern with different energy consumption values under a given frequency ratio. All the parabolas have a raft of points lower than baseline E_p , meaning that servers can save energy to some extent.

In what follows, we introduce in Section 6.2 a concept of *energy-saving window* to represent a portion of parabola that is lower than baseline E_p . We demonstrate how to apply our model to discover conditions under which servers may benefit from DVFS. We discuss in Section 6.3 that the energy efficiency of servers largely depends on λ .

6.2. Energy-saving windows

Now we introduce energy-saving windows, within which frequency ratios lead to energy savings using DVFS. In other words, frequency ratios outside an energy-saving window are unable to take leverage DVFS to conserve energy. We refer to a frequency ratio within an energy-saving window as *energy-saving frequency ratio*.

Fig. 5 shows an example of energy-saving windows and energy-saving frequency ratios when static power proportion is set to 20%. Recall that when frequency ratio r equals to 1 (i.e., $r = 1$, $f = 100\%$

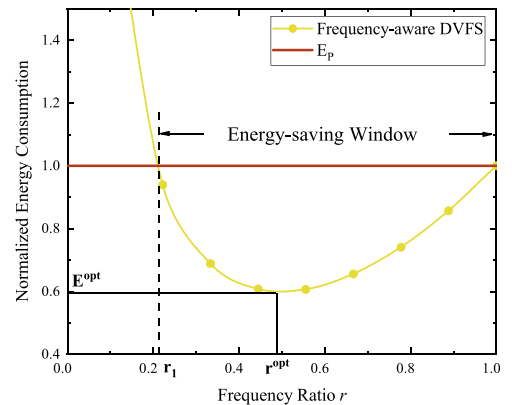


Fig. 5. An example of energy-saving windows and optimal energy-saving frequent ratio.

f^{max}), energy consumption spikes at the value of E_p , which is expressed as

$$E_p = \frac{\sum_{i \in T} \Gamma_i}{f^{max}} (P^{sta} + P^{dmax}). \quad (21)$$

Fig. 5 suggests that the curve obtained from our model may go beyond the red line (i.e., from 0 to f_{r1}), meaning that there exist cases where DVFS fails in offering energy savings. This phenomenon is expected, because the DVFS technology reduces frequency and voltage to save dynamic energy at the cost of increasing static power consumption. When frequency ratios are set to a value that is outside the energy-saving window, a large amount of energy is consumed during excessively enlarged execution times. On the flip side, the curve below the red line in Fig. 5 represents energy consumption when frequency ratios are in the range of the energy-saving window (i.e., r_1 to 1).

Fig. 5 intuitively shows that the upper bound of an energy-saving window is 1 and the lower bound is r_1 . Let us represent energy consumption obtained from our model as E_{DVFS} rather than E_c for processor c . Because peak energy consumption E_p (see (21)) is larger than E_{DVFS} inside the energy-saving window, we express the correlation between E_{DVFS} and E_p in (23) by incorporating (14). In light of (23) we show that frequency ratio r is greater than $\frac{\sqrt{1 + 4P^{sta}/P^{dmax}} - 1}{2}$, signifying the lower bound of an energy-saving window (see also r_1 in Fig. 6).

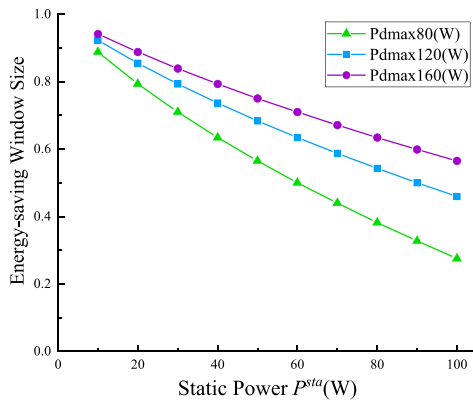
$$E_p - E_{DVFS} = \frac{\sum_{i \in T} \Gamma_i}{f^{max}} (P^{sta} + P^{dmax}) - \frac{\sum_{i \in T} \Gamma_i}{f^{max}} \left(\frac{P^{sta}}{r} + P^{dmax}(r)^2 \right) > 0 \quad (22)$$

$$r > \frac{\sqrt{1 + 4P^{sta}/P^{dmax}} - 1}{2} \quad (23)$$

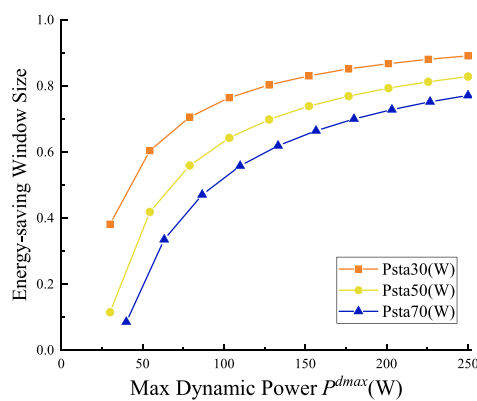
Given the upper and lower bounds of an energy-saving window, we argue that frequent ratio r residing within the energy-saving window offers energy savings. In other words, energy can be conserved by DVFS

$$\text{if we have } r \in \left(\frac{\sqrt{1 + 4P^{sta}/P^{dmax}} - 1}{2}, 1 \right).$$

Eq. (23) suggests that an energy-saving window is dependent of server static power P^{sta} and maximum dynamic power P^{dmax} . To further explore intriguing trends of energy-saving windows, we plot in Fig. 6 energy window size as functions of static power and maximum dynamic power. We shine a bright light on energy-saving window size, because a large window size implies there are ample opportunities to conserve energy using DVFS.



(a) Static Power vs. Energy-Saving Window Size



(b) Maximum Dynamic Power vs. Energy-Saving Window Size

Fig. 6. Impacts of static power and maximum dynamic power on the energy-saving window size in various server settings.

Fig. 6a shows the energy-saving window size of servers when the maximum dynamic power settings are 80 W, 120 W, 160 W, respectively. Similarly, Fig. 6b plots the energy-saving window size as functions of maximum dynamic power when static power is configured to 30 W, 50 W, 70 W, respectively. We observe from Fig. 6a that regardless of the maximum dynamic power, increasing static power continuously curtails the energy-saving window size. On the contrary, Fig. 6b reveals that energy-saving window size goes up with an rising maximum dynamic power. Interestingly, energy-saving window size is quite sensitive to maximum dynamic power when the dynamic value is below 120 W; when maximum dynamic power is sitting at a high level (e.g., 200 W), further pushing up the maximum dynamic power leads to a marginal growth in energy-saving window size. In summary, the energy-saving window size is proportional to maximum dynamic power and inversely proportional to static power.

6.3. Optimal energy-saving frequency ratio

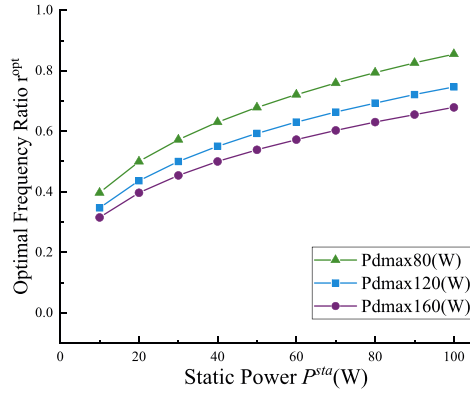
Fig. 5 unravels normalized energy consumption as a function of frequency ratio r . We mark the energy-saving window (i.e., from r_1 to 1) in Fig. 5, where an optimal frequency ratio (see r^{opt}) is the lowest point of the energy-consumption value obtained from our frequency-aware DVFS model. It is noteworthy that frequency ratio r^{opt} minimizes energy consumption denoted as E^{opt} . The discovery of the r^{opt} value is the centerpiece of our algorithm, because energy consumption can be minimized by adjusting frequency ratio to reach r^{opt} .

Eq. (17) suggests that the optimal frequency ratio is affected by static power and maximum dynamic power. As such, we plot optimal frequency ratio as functions of static power and maximum dynamic power in Fig. 7.

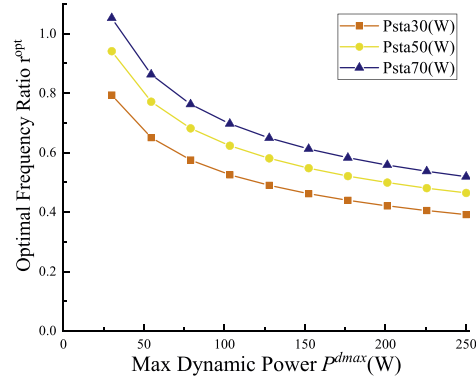
Similar to Fig. 6, Fig. 7 plots the optimal frequency ratio as the functions of static power and maximum dynamic power, respectively. Comparing the five parabolas in Fig. 4, we conclude that E^{opt} climbs with an increasing value of r^{opt} . Moreover, a high E^{opt} value leads to low energy-saving performance. Fig. 7a reveals that the optimal energy-saving frequency ratio is proportional to the static power, meaning that a large static power gives rise to a small amount of saved energy. On the flip side, Fig. 7b shows that the optimal frequency ratio is inversely proportional to the maximum dynamic power when the static power is a constant.

6.4. Static power proportion

Recall that energy-saving windows and optimal energy-saving frequency ratios are dependent of static power and maximum dynamic



(a) Static Power vs. Optimal Frequency Ratio



(b) Maximum Dynamic Power vs. Optimal Frequency Ratio

Fig. 7. Impacts of static power and maximum dynamic power on the optimal frequency ratio in various server settings.

power. To simplify the presentations of the results plotted in Figs. 6 and 7, we adopt a concept of static power proportion (i.e., λ) (see the second paragraph in Section 2.2). Fig. 8 intuitively illustrates the impacts of static power proportion λ on energy-saving windows and optimal frequency ratios when the static power proportion varies from 10% to 50%, which resembles the static power proportions of most modern servers.

Fig. 8 unravels that a small static power proportion of a server leads to a large energy-saving window size and a low optimal energy-saving frequency ratio. From the findings discussed in Section 6.3, we observe that a low r^{opt} value implies good energy-saving performance. On the other hand, a large energy-saving window size means there are ample opportunities to conserve energy using DVFS. In contrast, when the static power proportion λ grows, the energy-saving window size is narrowed coupled with an increasing r^{opt} value. We conclude that servers with large static power proportions tend to deliver deteriorated energy-saving performance.

In summary, our energy-saving model is evident that servers that have small λ parameters (a.k.a., static power proportion) are capable of taking advantages from the DVFS technique to offer high energy efficiency. This trend is reasonable, because a small static power proportion provide a wide space to tune frequencies to harvest energy savings. On the other hand, DVFS is not an ideal technique to conserve energy for servers where the λ parameters are high.

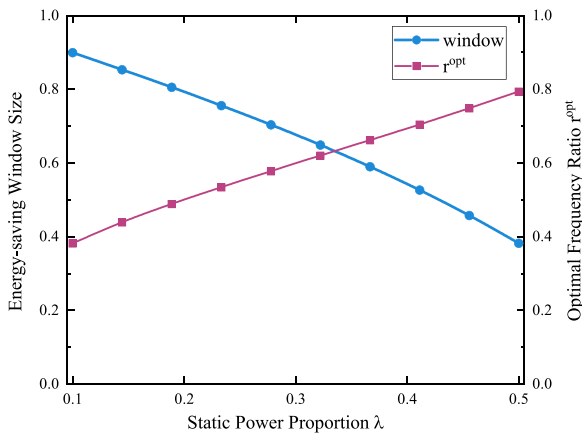


Fig. 8. Impacts of static power proportion λ on energy-saving windows and optimal frequency ratios.

6.5. Power consumption compared with baseline

We compare the energy consumption of servers equipped with and without our frequency-aware DVFS technique. The energy efficiency of the non-DVFS enabled system is represented by peak energy consumption E_p , which is a baseline measure offering no energy savings. In this group of experiments, the CPU static power is 100 W; the workload is configured to 10,000 clock cycles. Fig. 9 shows the energy savings obtained by the frequency-aware DVFS scheme under various static power proportion values. The findings confirm that energy savings offered by frequency-aware DVFS becomes pronounced under low static power proportion; the energy-efficiency strength of DVFS is diminishing when the static power proportion grows. This trend is consistent with those discussed in Section 6.4.

7. A numerical study and discussions

We conduct simulations to demonstrate the usage of the frequency-aware DVFS model elaborated in Section 5. More specifically, we simulate the energy consumption behaviors of three servers governed by the power management policies.

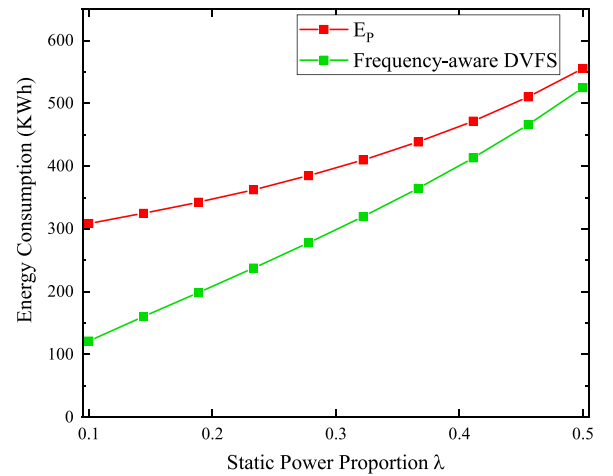


Fig. 9. An energy consumption comparison between servers equipped with and without the frequency-aware DVFS technique. E_p is the peak energy consumption of the non-DVFS enabled system. Static power proportion λ is configured in a range between 0.1 and 0.5.

7.1. Experimental setup

As a numerical study, we implement the model to test five popular CPUs, namely, Xeon E5-2670 [39,41], Intel i7-4770 [42,43], AMD Athlon, Intel Pentium 950, and Intel Pentium 4630 [40]. We make use of the static power p^{sta} , maximum dynamic power p^{dmax} , and the frequency levels of the five real-world processors to demonstrate a way of applying our model to determine the most energy-efficient frequency for real-world settings. Table 2 summarizes the three parameters of the five CPUs.

We implement the proposed mathematical model, the input of which includes a static power and the max dynamic power. The output of the model implementation is an optimal frequency ratio accompanied by the lowest energy consumption. To illustrate the usage of our implemented model, we compare our frequency-aware DVFS with the two competitive approaches in the context of the five tested processors. After deriving the optimal frequency ratios coupled with the minimized energy consumption, we put spotlight on the energy-efficiency comparison of our frequency-aware DVFS scheme against the utilization-based DVFS scheme and a baseline scheme. Let us briefly articulate the key ideas of the three schemes below.

- **The baseline scheme.** In this scenario, the simulated servers are operated at the maximum frequency. The energy consumption of the servers managed by the baseline scheme is referred to as peak energy consumption E_p (see also (21) in Section 6.2).
- **The utilization-based DVFS scheme [16,44].** This popular DVFS scheme adjusts CPU frequency according to processor utilization that is dynamically changing. A server's power consumption is proportional to the server's utilization [16]. The correlation between power consumption and CPU utilization is expressed as

$$P = P_{idle} + (P_{busy} - P_{idle}) \cdot u \quad (24)$$

where P_{idle} and P_{busy} are power consumption of a system when its CPU's power state is idle and busy, respectively; u is the CPU utilization. In prior studies (see, for example, [44]), the optimal utilization in the utilization-based DVFS scheme is set to 80%. Please refer to [44] for the detailed description on the utilization-based DVFS scheme.

- **The frequency-aware DVFS scheme.** Our proposed energy-saving method explores frequency ratio to optimize energy efficiency. Obtaining r^{opt} from Eq. (17), our scheme minimize energy consumption by electing a frequency-ratio value r that is close to r^{opt} .

7.2. Experimental results

Fig. 10 shows the results of normalized energy consumption of the AMD and Intel processors governed by three power management policies (i.e., frequency-aware DVFS, utilization-based DVFS scheme, baseline). The normalized energy consumption of the baseline approach is fixed at the value of 1.0. Therefore, the baseline's energy consumption is marked as a red line rather than multiple bars. Compared with the baseline policy, our frequency-aware DVFS conserves the energy by anywhere between 19% and 41%. The energy savings measured by the simulator are consistent with estimates offered by the model elaborated

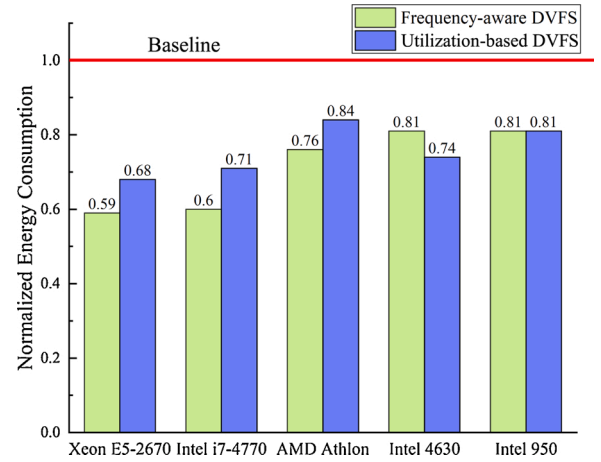


Fig. 10. Normalized energy consumption of the AMD and Intel processors managed by the frequency-aware DVFS scheme, utilization-based DVFS scheme, and the baseline scheme.

in Section 6.5. The simulation results validate the correctness of the model developed in this study.

Comparing the energy efficiency between our frequency-aware DVFS method and the utilization-based DVFS technique, we observe three intriguing and surprising trends. When it comes to the Xeon E5-2670, Intel i7-4770, and the AMD Athlon, the frequency-aware DVFS technique conserves more energy than the utilization-based DVFS counterpart. For instance, compared with utilization-based DVFS, frequency-aware DVFS slashes the energy consumption of Intel i7-4770 by 15.5%. From the perspective of the Intel Pentium 4630 processor, the utilization-based DVFS method is superior to the frequency-aware DVFS policy. In the context of the Intel Pentium 950 processor, the frequency-aware and utilization-based DVFS schemes share identical energy efficiency. Such versatile trends motivate us to shed some light on the applicability of the frequency-aware DVFS in the next subsection.

7.3. Applicability discussions

The results plotted in Fig. 10 suggest that frequency-aware DVFS outperforms the utilization-based DVFS scheme on three cases (i.e., Xeon, Intel i7, and AMD Athlon), whereas the opposite is true for an outlier – Intel 4630. We delve into the features of these five processors to justify the three winning cases of Xeon, Intel i7, and AMD Athlon, in which the frequency-aware DVFS scheme more energy efficient than the utilization-based alternative.

The frequency-aware scheme is not on par with the utilization-based technique in the case of Intel Pentium 4630, because this Intel processor has no appropriate frequency levels that take full advantage from for our scheme. For example, optimal frequency ratio r^{opt} of Intel Pentium 4630 is 0.617; unfortunately, the nearest ratio close to r^{opt} is just 0.5, which is far away from the optimal energy-saving frequency ratio of 0.617. When it comes to Intel Pentium 950, the frequency-based and utilization based schemes are equally energy efficient. This trend is reasonable, because, Pentium 950's static power proportion λ is over 35% – the highest one among those of the three processors. Recall that (see Section 6.4) a high λ value suppresses the capability of DVFS to conserve energy by adjusting frequency, adversely affecting the energy-saving performance.

The aforementioned comparisons of the policies managing the five processors demonstrate that our model is adroit at analyzing the energy efficiency of DVFS-enabled servers. The results suggest that servers powered by processors with more frequency levels have a higher likelihood to achieve the lowest energy consumption, because the frequency ratios of these processors are more likely to be closer to optimal ratio r_{opt} . Furthermore, servers with low static power proportion can enjoy energy savings offered by our frequency-aware DVFS model.

Table 2

The CPU configurations of the three tested servers.

Server	Frequency (GHz)	p^{sta} (W)	p^{dmax} (W)
Xeon E5-2670	{1.2,1.3,...,2.5,2.6}	84.1	342.9
Intel i7-4770	{0.8,1.0,...,3.2,3.4}	19	76
AMD Athlon	{0.8,1.5,2.0,2.7,3.0}	53	121.1
Intel Pentium 950	{1.0,1.5,2.2,2.9,3.4}	68	126
Intel Pentium 4630	{1.0,1.2,1.5,2.4,3.0}	30	64

Our frequency-aware DVFS method delivers comparable energy-saving performance as the utilization-based DVFS policy. A distinctive feature of our model is that frequency ratio is introduced to measure the QoS and energy consumption. The existing utilization-based DVFS model [44] assumes that virtual machine execution times are given in a priori; more often than not, the estimates of the execution times are inaccurate. In contrast, our frequency-aware model applies DVFS to conserve energy without estimating virtual machine execution times. Our model makes energy-saving decisions in accordance with server configuration data. Our model is more practical than the existing utilization-based model.

8. Conclusions and future work

In this study, we have developed a frequency-aware DVFS model for virtual machines running on cloud computing platforms. It is conventional wisdom to apply worst-case execution time or WCET as QoS requirements of virtual machines running on DVFS-enabled systems, which conserve energy by scaling down CPU frequency. Our model advocates for specifying QoS requirements in form of frequency rather than WCET. The prominent benefit of our proposed frequency requirements is to avert inaccurate WCET estimates. We demonstrated the feasibility of transforming the traditional time-aware QoS model into a novel frequency-aware QoS model. We showed that our model can be applied to govern the power management of DVFS-enabled systems without violating the SLAs of virtual machines. In the model, we introduced a new parameter referred to as *frequency ratio*, which impacts the energy consumption of processors.

Given a processor's hardware information, our model is adept at deriving the optimal frequency ratio that leads to the minimum energy consumption of the processor. A power manager can maximize energy efficiency by adjusting the processor's frequency to meet the optimal frequency ratio. After analyzing the correlations between frequency ratio and energy consumption, we drew an intriguing conclusion – a small static power proportion leads to high energy-saving performance. To shed bright light on the applicability of our frequency-aware DVFS model, we conducted a simulation study using the parameters from three real-world processors. We confirmed that the energy-efficiency performance of our model is on par with the well-known utilization-based DVFS scheme. The simulation results are evident that our model is conducive to projecting the energy-saving performance of DVFS-enabled computing systems running virtual machines.

This study lays out a solid theoretical foundation for power management software tailored for DVFS-enabled clouds. There are three future research directions that will further expand this work. First, we intend to dabble a little bit in the development of a power manager where the frequency-aware model is incorporated to adjust CPU frequencies in a heterogeneous computing environment on clouds. The frequency-aware model will offer the power manager insightful hints on optimal frequency ratios, which boost the energy efficiency of heterogeneous processors. Second, we have a solid plan to dive into the development of a hybrid mechanism that seamlessly blends the DVFS and VM consolidation techniques. Third, thanks to the prominent energy-saving features offered by GPU DVFS, we plan to explore energy conservation techniques in GPU computing systems. We will forge our model to guide the configuration of frequency ratios minimizing GPU's energy consumption. We also have a desire to push our future directions further by upgrading the model to handle CPU-GPU heterogeneous systems in the face of cloud computing.

Declaration of Competing Interest

The authors report no declarations of interest.

References

- [1] D. Reinsel, J. Gantz, J. Rydning, *The Digitization of the World: From Edge to Core*, International Data Corporation, Framingham, 2018.
- [2] Data Center Market – Global Outlook and Forecast 2018–2023, Tech. rep., Arizton 4577457, 2018.
- [3] G. IDEX, Cisco Global Cloud Index: Forecast and Methodology, 2016–2021, Şubat, Cisco, San Jose, CA, USA, 2018. White Paper 1513879861264127.
- [4] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, H. Jiang, On arbitrating the power-performance tradeoff in SaaS clouds, *IEEE Trans. Parallel Distrib. Syst.* 25 (10) (2013) 2648–2658.
- [5] I.T.R. Group, et al., Top 10 Energy-Saving Tips for a Greener Data Center, 2007.
- [6] L.A. Barroso, J. Clidaras, U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 2nd ed., 2013 doi:10.2200/S00516ED2V01Y201306CAC024.
- [7] L.A. Barroso, U. Hölzle, *The Case for Energy-Proportional Computing*, 2007.
- [8] J. Mei, K. Li, A. Ouyang, K. Li, A profit maximization scheme with guaranteed quality of service in cloud computing, *IEEE Trans. Comput.* 64 (11) (2015) 3064–3078.
- [9] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, B. Li, iAware: making live migration of virtual machines interference-aware in the cloud, *IEEE Trans. Comput.* 63 (12) (2013) 3012–3025.
- [10] W. Jiang, F. Liu, G. Tang, K. Wu, H. Jin, Virtual machine power accounting with Shapley value, 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE (2017) 1683–1693.
- [11] M.P. Papazoglou, Service-oriented computing: concepts, characteristics and directions, in: *Proceedings of the Fourth International Conference on Web Information Systems Engineering*, 2003, WISE 2003, IEEE, 2003, pp. 3–12.
- [12] Q. Wu, Q. Deng, L. Ganesh, C.-H. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, Y.J. Song, Dynamo: Facebook's data center-wide power management system, 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), IEEE (2016) 469–480.
- [13] S. Roy, A. Rudra, A. Verma, An energy complexity model for algorithms, *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ACM (2013) 283–304.
- [14] B.M. Tudor, Y.M. Teo, On understanding the energy consumption of ARM-based multicore servers, *ACM SIGMETRICS Perform. Eval. Rev.* 41 (1) (2013) 267–278.
- [15] E.M. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, in: *International Workshop on Power-Aware Computer Systems*, Springer, 2002, pp. 179–197.
- [16] X. Fan, W.-D. Weber, L.A. Barroso, Power provisioning for a warehouse-sized computer, in: *ACM SIGARCH Computer Architecture News*, vol. 35, ACM, 2007, pp. 13–23.
- [17] Y. Gao, H. Guan, Z. Qi, B. Wang, L. Liu, Quality of service aware power management for virtualized data centers, *J. Syst. Architect.* 59 (4-5) (2013) 245–259.
- [18] G. Tang, W. Jiang, Z. Xu, F. Liu, K. Wu, NIPD: non-intrusive power disaggregation in legacy datacenters, *IEEE Trans. Comput.* 66 (2) (2016) 312–325.
- [19] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu, H. Jin, Carbon-aware online control of geo-distributed cloud services, *IEEE Trans. Parallel Distrib. Syst.* 27 (9) (2015) 2506–2519.
- [20] F. Hao, M. Kodialam, T. Lakshman, S. Mukherjee, Online allocation of virtual machines in a distributed cloud, *IEEE/ACM Trans. Network.* 25 (1) (2017) 238–249.
- [21] K.K. Nguyen, M. Cheriet, Environment-aware virtual slice provisioning in green cloud environment, *IEEE Trans. Serv. Comput.* 8 (3) (2014) 507–519.
- [22] F. Yao, A. Demers, S. Shenker, A scheduling model for reduced CPU energy, *Proceedings of IEEE 36th Annual Foundations of Computer Science*, IEEE (1995) 374–382.
- [23] H. Aydin, Q. Yang, Energy-aware partitioning for multiprocessor real-time systems, *Proceedings International Parallel and Distributed Processing Symposium*, IEEE (2003) 9.
- [24] C.-M. Wu, R.-S. Chang, H.-Y. Chan, A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters, *Future Gen. Comput. Syst.* 37 (2014) 141–147.
- [25] K. Li, Design and analysis of heuristic algorithms for power-aware scheduling of precedence constrained tasks, 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, IEEE (2011) 804–813.
- [26] L. Wang, G. Von Laszewski, J. Dayal, F. Wang, Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS, in: *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, IEEE Computer Society, 2010, pp. 368–377.
- [27] C.-Y. Yang, J.-J. Chen, T.-W. Kuo, An approximation algorithm for energy-efficient scheduling on a chip multiprocessor, *Design, Automation and Test in Europe*, IEEE (2005) 468–473.
- [28] F. Kong, W. Yi, Q. Deng, Energy-efficient scheduling of real-time tasks on cluster-based multicores, 2011 Design, Automation & Test in Europe, IEEE (2011) 1–6.
- [29] X. Ruan, X. Qin, Z. Zong, K. Bellam, M. Nijim, An energy-efficient scheduling algorithm using dynamic voltage scaling for parallel applications on clusters, 2007 16th International Conference on Computer Communications and Networks, IEEE (2007) 735–740.
- [30] Z. Zong, X. Qin, X. Ruan, K. Bellam, M. Nijim, M. Alghamdi, Energy-efficient scheduling for parallel applications running on heterogeneous clusters, 2007 International Conference on Parallel Processing (ICPP 2007), IEEE, 19–19 (2007).

- [31] X. Qin, M. Alghamdi, M. Nijim, Z. Zong, K. Bellam, Scheduling of periodic packets in energy-aware wireless networks, 2007 IEEE International Performance, Computing, and Communications Conference, IEEE (2007) 210–217.
- [32] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, et al., The worst-case execution-time problem-overview of methods and survey of tools, *ACM Trans. Embedded Comput. Syst.* 7 (3) (2008) 36.
- [33] K.H. Kim, R. Buyya, J. Kim, Power aware scheduling of bag-of-tasks applications with deadline constraints on DVS-enabled clusters, Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07), IEEE (2007) 541–548.
- [34] S. Zhuravlev, J.C. Saez, S. Blagodurov, A. Fedorova, M. Prieto, Survey of energy-cognizant scheduling techniques, *IEEE Trans. Parallel Distrib. Syst.* 24 (7) (2012) 1447–1464.
- [35] T.D. Burd, R.W. Brodersen, Energy efficient CMOS microprocessor design, *Proceedings of the Twenty-Eighth Annual Hawaii International Conference on System Sciences*, vol. 1, IEEE (1995) 288–297.
- [36] S. Eyerhan, L. Eeckhout, Fine-grained DVFS using on-chip regulators, *ACM Trans. Architect. Code Optim.* 8 (1) (2011) 1–24.
- [37] X. Wang, M. Chen, Cluster-level feedback power control for performance optimization, 2008 IEEE 14th International Symposium on High Performance Computer Architecture, IEEE (2008) 101–110.
- [38] D. Li, J. Wu, Energy-aware scheduling for aperiodic tasks on multi-core processors, 2014 43rd International Conference on Parallel Processing, IEEE (2014) 361–370.
- [39] P. Alcorn, Intel Xeon e5-2600 v4 Broadwell-ep Review, 2016. <https://www.tomshardware.com/reviews/intel-xeon-e5-2600-v4-broadwell-ep,4514-8.html>.
- [40] J. Li, Z. Li, K. Ren, X. Liu, Towards optimal electric demand management for internet data centers, *IEEE Trans. Smart Grid* 3 (1) (2011) 183–192.
- [41] A. Quintiliani, M. Chinnici, D. De Chiara, Understanding “workload-related” metrics for energy efficiency in data center, in: 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), IEEE, 2016, pp. 830–837.
- [42] OpenBenchmarking, CPUfreq-info – Intel core i7-4770, <https://openbenchmarking.org/system/1408184-KH-HETZNEREX64/Intel%20Core%20i7-4770/cpufreq-info>.
- [43] Servethehome, Intel Core i7-4770 Power Consumption, <https://www.servethehome.com/intel-core-i7-4770-review-benchmarks-haswell-retail-unleashed/intel-core-i7-4770-power-consumption/>.
- [44] P. Arroba, J.M. Moya, J.L. Ayala, R. Buyya, Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers, *Concurrency Comput.: Pract. Exp.* 29 (10) (2017) e4067.