2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC) | 978-1-7281-9829-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/IPCCC50635.2020.9391552

Modeling Energy Consumption of Virtual Machines in DVFS-Enabled Cloud Data Centers

Jianzhou Mao, Tathagata Bhattacharya, Xiaopu Peng,
Ting Cao, and Xiao Qin
Department of Computer Science and Software Engineering
Auburn University, Auburn, AL.

***Abstract*—To cut back energy consumption of virtual-machine-powered data centers, we build an optimization model for virtual machines running in DVFS-enabled cloud data centers. With the model in place, cloud computing systems are equipped to keep track of dynamic power and static power of processors in virtual machines. Unlike existing dynamic voltage and frequency scaling schemes, our solution orchestrates frequency requirements rather than task execution times. The model makes it possible to obtain an optimal frequency ratio, which minimizes energy consumption of virtual machines. As a result, a data center's energy efficiency is boosted by controlling CPU frequency to meet the optimal frequency ratio. We demonstrate a way of manipulating frequency ratios to pushing up energy efficiency without violating virtual machines' frequency requirements. The experimental results unveil that our modeling approach offers a practical way of conserving the energy consumption of virtual machines running in data centers.**

***Index Terms*—cloud computing, DVFS, energy optimization, static power proportion,virtual machines, frequency ratio.**

I. Introduction

The performance of cloud computing platforms has been significantly growing in past decades. With the dramatic evolution in computing capacity, the power consumption of data centers 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 data centers.

Our frequency-aware scheduler aims to minimize CPU energy cost in virtual-machine-enabled data centers 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 data centers in the future.
- Large-scale data centers 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.

Servers, core IT components in data centers, are major players contributing to high energy cost (e.g., more than 26%) [1]. The energy cost of CPU, in turn, occupies the highest proportion in the server (i.e., 42%) [2]; in some scenarios, CPUs account for up to 70% of total energy consumption in servers [3]. Thus, conserving CPU energy is indispensable in tackling the data center energy problem. The dynamic voltage and frequency scaling technique or DVFS is one of the most widely adopted schemes to reduce CPU power; DVFS enables processors consume less power by adjust 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 DVFS-endabled data centers.

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 orchestrate 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 experiment to evaluate energy efficiency of servers running multiple virtual machines in data centers.

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

- We introduce the concept of frequency ratio to quantify relative CPU frequency under various conditions.
- We design a QoS-base strategy that adjusts frequency ratios to optimize energy efficiency.
- We propose a model that seamlessly integrates static and dynamic power consumption.
- We apply our proposed model to measure the energy efficiency of virtual-machine-powered servers in data centers.

The remainder of this paper is organized as follows. Sec-

978-1-7281-9829-3/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:10:51 UTC from IEEE Xplore. Restrictions apply.

tion II outlines the previous studies related to energy conservation techniques, DVFS, and task models. Section III 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 Section IV and Section V, respectively. The analysis of the DVFS model with respect to frequency ratio can be found in Section VI. Section VII discusses the sample usages and applicability of the proposed frequency-aware model. We conclude this paper in Section VIII.

II. Related Work

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

A. Energy Conservation Techniques

In the past decade, a growing number of energy-saving methods have been devised to conserve the energy consumption in data centers [4] [5]. For example, Hao *et al.* [6] proposed a real-time virtual machine placement algorithm customized for distributed data centers. This placement algorithm maximizes service capabilities by adjusting virtual machine deployment, thereby enabling data centers to serve more users with short response times. 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 [7]. Aydin and Yang [8] 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.* [9] 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.

B. 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 [10]. For instance, Wang *et al.* [11] 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 [12] [13]; 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 [8]. With the tasks models in place, numerous scheduling algorithms were implemented for saving energy of computing systems running real-time applications [14] [15] [16]. 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.

III. Frequency-Aware QoS

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.

In real-time task modeling, it is a conventional wisdom to adopt WCET (i.e., Worst case excution-time) and deadlines (i.e., upper timing bound) to characterize QoS requirements of real-time applications [17]. 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. 1 depicts two types of real-time tasks coexisting in our frequency-aware QoS modeling. For the traditional real-time tasks (see the left-hand side of Fig. 1), real-time tasks should entail timing 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 right-hand side of Fig. 1), non-time-sensitive applications merely pass on frequency requirements to the DVFS-enabled system.

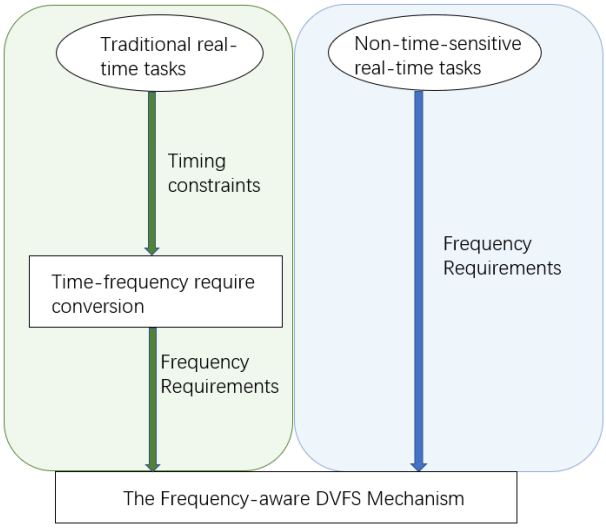


Fig. 1. frequency-aware modeling for time-sensitive and non-time-sensitive real-time tasks running in DVFS enabled systems.

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.

In real-world scenarios, there might be the lack of specified requirements in terms of deadline or WCET in submitted tasks.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:10:51 UTC from IEEE Xplore. Restrictions apply.

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. 1). 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.

IV. 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} [18]:

$$E = E^{sta} + E^{dyn}. \tag{1}$$

Prior studies [19] [12] [20] confirmed that dynamic power P^{dyn} is computed as

$$P^{dyn} = C_l \cdot A \cdot V_{dd}^2 \cdot f, \tag{2}$$

where C_l is load capacitance, A is the percentage of active gates, V_{dd} is supply voltage, and f is processor frequency.

The correlation between frequency and voltage is expressed as:

$$f = k \cdot \frac{(V_{dd} - V_{th})^2}{V_{dd}}, \tag{3}$$

where V_{th} - typically a small value - is a threshold voltage [21] [22]. Approximately, there is a linear relation between frequency and voltage (i.e., $f \approx k \cdot V_{dd}$). Because Cl, 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

$$P^{dyn} = \beta \cdot f^3. \tag{4}$$

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 (5) from (4) as

$$P^{dyn} = P^{dmax} \cdot \frac{f^3}{(f^{dmax})^3}. \tag{5}$$

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

$$P = P^{sta} + P^{dmax} \cdot \frac{f^3}{(f^{max})^3}. \tag{6}$$

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 $10\%f^{max}$, $20\%f^{max}$,$40\%f^{max}$, $100\%f^{max}$ [23]. 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 $r = f/f^{max}$. The total CPU power can obtained from frequency ratio r as

$$P = P^{sta} + P^{dmax} \cdot r^3. \tag{7}$$

V. Modeling Frequency-aware DVFS

Now we are in a position to present a DVFS model tailored for tasks issuing frequency requirements (see Fig. 1). 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

$$E_i = P_i \cdot t_i = P_i \cdot \frac{\Gamma_i}{f_i}. \tag{8}$$

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 jth segment ($1 \leq j \leq K_i$) is $\gamma_{i,j}$. Thus, the clock cycles of all the segments of virtual machine vm_i form a frequency set $\{\gamma_{i,1}, \gamma_{i,2}, ..., \gamma_{i,K_i}\}$, which determines the total execution requirements Γ_i of virtual machine vm_i. Thus, we have

$$\Gamma_i = \sum_{j=1}^{K_i} \gamma_{i,j}. \tag{9}$$

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

$$E_i = \sum_{j=1}^{k} P_{i,j} \frac{\gamma_{i,j}}{f_{i,j}}. \tag{10}$$

The *Lagrange* multiplier method is applied to minimize energy E_i [24]. In case where frequencies of vm_i during all its intervals are identical, the computation of (10) 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

$$f_i = f_{i,1} = f_{i,2} = f_{i,3} = ... = f_{i,K}. \tag{11}$$

Hence, the optimal frequency of each segment is equal to the optimal solution for the entire virtual machine vm_i. Combining with the energy formula (see (8)), we derive energy consumption of processor c as

$$E_c = \frac{\sum_{i \in T} \Gamma_i}{f_c^{max}} \left(\frac{P_c^{sta}}{r_c} + P_c^{dmax} (r_c)^2\right). \tag{12}$$

(12) 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

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:10:51 UTC from IEEE Xplore. Restrictions apply.

$$E'_c = \frac{\sum_{i \in T} \Gamma_i}{f_c^{max}} \cdot \frac{1}{(r_c)^2}(2 \cdot (r_c)^3 \cdot P_c^{dmax} - P_c^{sta}). \tag{13}$$

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

$$r_c^{opt} = \sqrt[3]{\frac{P_c^{sta}}{2P_c^{dmax}}}. \tag{14}$$

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.

$$E_c^{opt} = \frac{\sum_{i \in T} \Gamma_i}{f_c^{max}}(\frac{P_c^{sta}}{r_c^{opt}} + P_c^{dmax} \cdot (r_c^{opt})^2). \tag{15}$$

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., $C = \{c_1, ..., c_m\}$) in a data center. 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:

$$E_C^{opt} = \sum_{c \in C} E_c^{opt}. \tag{16}$$

where E_c^{opt} on the right-hand side of the expression is obtained from (15).

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.

VI. Analysis of Frequency Ratios

Recall that in our proposed frequency-aware DVFS model (see Section V), optimal energy consumption derived from Eq. (15) 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.

Eq. (12) indicates the correlation between energy consumption and frequency ratio r. Fig. 2 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. 2 broadly representative, we introduce a parameter called *static power proportion* in lieu of a server's power data. Eq. (12) 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 $\lambda = P^{sta} / (P^{dmax} + P^{sta})$. More often than not, a server's static power proportion λ is anywhere between 15% and 35 %. 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.

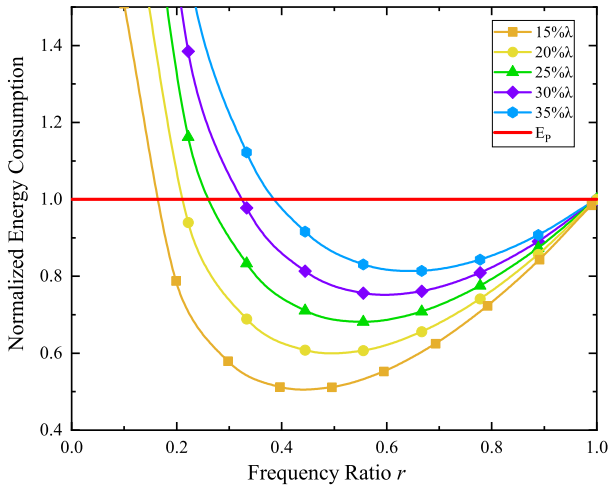


Fig. 2. Impacts of frequency ratio on normalized energy consumption under various value.

Fig. 2 shows five parabolas of the function expressed in Eq. (12) 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. 2 are the normalized energy consumption when the static power proportion λ is configured to 15%, 20%, 25%, 30%, and 35%, respectively. Fig. 2 depicts the energy consumption outcomes of the five λ values that cover a majority of real-world server cases.

The five parabolas in Fig. 2 share a similar pattern with respect to energy consumption values under a given frequency ratio. All the parabolas have a raft of points lower than baseline E_p, meaning that the servers embrace ample opportunities to conserve energy. The findings unveil that static power proportion λ is proportional to the energy consumption. The

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:10:51 UTC from IEEE Xplore. Restrictions apply.

results confirm that a server with a low static power proportion λ has an energy-efficiency edge over its counterparts with high λ values.

VII. Simulation Results and Discussions

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

A. Experimental Setup

TABLE I. The CPU configurations of the three tested servers.

Server	Frequency (GHz)	$\mathrm{P}^{\mathrm{sta}}$ (W)	$\mathrm{P}^{\mathrm{dmax}}$ (W)
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 4360	{1.0,1.2,1.5,2.4,3.0}	30	64

We configure the tested servers in a way to simulate three popular CPUs, namely, AMD Athlon, Intel Pentium 950, and Intel Pentium 4360 [25]. Table I summarizes the system parameters of the three CPUs with respect to the available frequency levels, static power p^{sta}, and maximum dynamic power p^{dmax}.

We put spotlight on three scenarios to evaluate the performance of our frequency-aware DVFS scheme. For this purpose, we compare our frequency-aware DVFS with two competitive approaches. The first approach is treated as a baseline scheme, in which the simulated servers are operated at the maximum frequency. The second one is referred to as the utilization-based DVFS scheme [26] [27] - a popular DVFS scheme that adjusts CPU frequency according to dynamically changing processor utilization. In the prior studies (see, for example, [27]), the optimal utilization in the utilization-based DVFS scheme is set to 80%. Please refer to [27] for the detailed description on the utilization-based DVFS scheme.

B. Experimental Results

Fig. 3 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). Compared with the baseline policy, our frequency-aware DVFS conserves the energy by anywhere between 19% to 24%. 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 AMD server, the frequency-aware DVFS technique conserves more energy than the utilization-based DVFS counterpart. From the perspective of the Intel Pentium 4360 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.

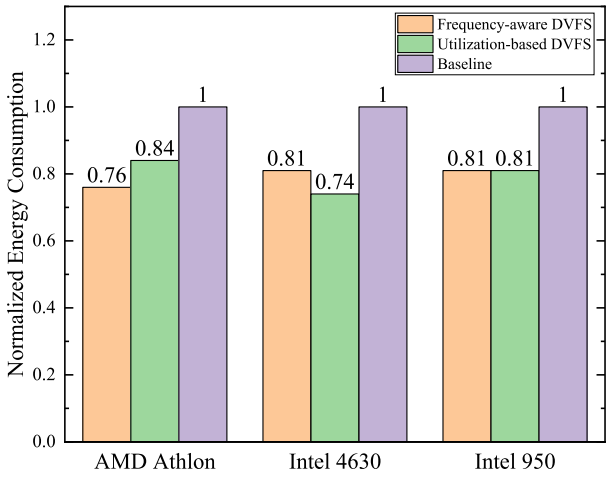


Fig. 3. Normalized energy consumption of the AMD and Intel processors managed by the frequency-aware DVFS scheme, utilization-based DVFS scheme, and the baseline scheme.

C. Applicability Discussions

The results plotted in Fig. 3 suggest that frequency-aware DVFS outperforms the utilization-based DVFS scheme on some processors (e.g., AMD) whereas the opposite is true for other processors (e.g., Intel 4360). We delve into the features of these processors to figure out why the AMD processor rather than the two Intel processors takes full advantage from the frequency-aware DVFS scheme.

The frequency-aware scheme is not on par with the utilization-based technique in the case of Intel Pentium 4360, 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 4360 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 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 three 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.

Our frequency-aware DVFS method delivers comparable

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:10:51 UTC from IEEE Xplore. Restrictions apply.

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 [27] 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.

VIII. Conclusions

In this study, we developed a frequency-aware DVFS model for virtual machines running on cloud computing platforms. Unlike the conventional wisdom, Our model advocates for QoS requirements in form of frequency rather than WCET. In the model, we introduced a new parameter referred to as *frequency ratio*, which impacts the energy consumption of processors. Given processor hardware information, our model is adept at deriving optimal frequency ratios that minimum energy consumption of the processors. The simulation results are evident that our model is conducive to projecting the energy-saving performance of DVFS-enabled computing systems running virtual machines.

Acknowledgments

This work is supported by the U.S. National Science Foundation under Grants IIS-1618669 and OAC-1642133.

References

[1] I.-T. R. Group *et al.*, "Top 10 energy-saving tips for a greener data center," 2007.

[2] L. A. Barroso, J. Clidaras, and U. Hlzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*, 2013. [Online]. Available: http://dx.doi.org/10.2200/S00516ED2V01Y201306CAC024

[3] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," 2007.

[4] G. Tang, W. Jiang, Z. Xu, F. Liu, and K. Wu, "Nipd: Non-intrusive power disaggregation in legacy datacenters," *IEEE Transactions on Computers*, vol. 66, no. 2, pp. 312–325, 2016.

[5] 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*, vol. 27, no. 9, pp. 2506–2519, 2015.

[6] F. Hao, M. Kodialam, T. Lakshman, and S. Mukherjee, "Online allocation of virtual machines in a distributed cloud," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 1, pp. 238–249, 2017.

[7] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced cpu energy," in *Proceedings of IEEE 36th annual foundations of computer science*. IEEE, 1995, pp. 374–382.

[8] H. Aydin and Q. Yang, "Energy-aware partitioning for multiprocessor real-time systems," in *Proceedings International Parallel and Distributed Processing Symposium*. IEEE, 2003, pp. 9–pp.

[9] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the dvfs technique for cloud datacenters," *Future Generation Computer Systems*, vol. 37, pp. 141–147, 2014.

[10] K. Li, "Design and analysis of heuristic algorithms for power-aware scheduling of precedence constrained tasks," in *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*. IEEE, 2011, pp. 804–813.

[11] L. Wang, G. Von Laszewski, J. Dayal, and 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.

[12] C.-Y. Yang, J.-J. Chen, and T.-W. Kuo, "An approximation algorithm for energy-efficient scheduling on a chip multiprocessor," in *Design, Automation and Test in Europe*. IEEE, 2005, pp. 468–473.

[13] F. Kong, W. Yi, and Q. Deng, "Energy-efficient scheduling of real-time tasks on cluster-based multicores," in *2011 Design, Automation & Test in Europe*. IEEE, 2011, pp. 1–6.

[14] X. Ruan, X. Qin, Z. Zong, K. Bellam, and M. Nijim, "An energy-efficient scheduling algorithm using dynamic voltage scaling for parallel applications on clusters," in *2007 16th International Conference on Computer Communications and Networks*. IEEE, 2007, pp. 735–740.

[15] Z. Zong, X. Qin, X. Ruan, K. Bellam, M. Nijim, and M. Alghamdi, "Energy-efficient scheduling for parallel applications running on heterogeneous clusters," in *2007 International Conference on Parallel Processing (ICPP 2007)*. IEEE, 2007, pp. 19–19.

[16] X. Qin, M. Alghamdi, M. Nijim, Z. Zong, and K. Bellam, "Scheduling of periodic packets in energy-aware wireless networks," in *2007 IEEE International Performance, Computing, and Communications Conference*. IEEE, 2007, pp. 210–217.

[17] 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 problemoverview of methods and survey of tools," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, no. 3, p. 36, 2008.

[18] K. H. Kim, R. Buyya, and J. Kim, "Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters," in *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07)*. IEEE, 2007, pp. 541–548.

[19] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto, "Survey of energy-cognizant scheduling techniques," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1447–1464, 2012.

[20] R. Ge, X. Feng, and K. W. Cameron, "Performance-constrained distributed dvs scheduling for scientific applications on power-aware clusters," in *SC'05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*. IEEE, 2005, pp. 34–34.

[21] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power cmos digital design," *IEICE Transactions on Electronics*, vol. 75, no. 4, pp. 371–382, 1992.

[22] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power optimization of variable-voltage core-based systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 12, pp. 1702–1714, 1999.

[23] X. Wang and M. Chen, "Cluster-level feedback power control for performance optimization," in *2008 IEEE 14th International Symposium on High Performance Computer Architecture*. IEEE, 2008, pp. 101–110.

[24] D. Li and J. Wu, "Energy-aware scheduling for aperiodic tasks on multi-core processors," in *2014 43rd International Conference on Parallel Processing*. IEEE, 2014, pp. 361–370.

[25] J. Li, Z. Li, K. Ren, and X. Liu, "Towards optimal electric demand management for internet data centers," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 183–192, 2011.

[26] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ACM SIGARCH computer architecture news*, vol. 35, no. 2. ACM, 2007, pp. 13–23.

[27] P. Arroba, J. M. Moya, J. L. Ayala, and R. Buyya, "Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, p. e4067, 2017.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:10:51 UTC from IEEE Xplore. Restrictions apply.