19

ENERGY-SAVING TECHNOLOGIES OF SERVERS IN DATA CENTERS

WEIWEI LIN[1], WENTAI WU[2] AND KEQIN LI[3]

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[2] Department of Computer Science, University of Warwick, Coventry, United Kingdom
[3] Department of Computer Science, State University of New York, New Paltz, New York, United States of America

19.1 INTRODUCTION

We are enjoying various kinds of online services that offer so much fun and convenience to our life—in a large part, these should be credited to those enterprise-owned data centers though we may never be conscious of where they are. Speaking more precisely, it is those metal "labors" racked row upon row working 24 hours a day that uphold every demand of us. When you upload the latest holiday photos to Facebook, there's a chance they'll end up stored in one of the tens of thousands of servers in Prineville, Oregon, a small town where the company has built three giant data centers and two more are in plan. Undoubtedly, data centers are making life better, but benefits always come at a cost—data centers are gobbling up our energy. Already, data centers use an estimated 200 terawatt hours (TWh) each year, while Anders Andrae, a specialist in sustainable Information and Communication Technology (ICT), forecasts that the energy demand of ICT will accelerate in the 2020s, approaching 9,000 TWh by 2030 among which data centers will take a large slice. With this trend, topics related to energy conservation have gradually become the focus when people are talking about data centers.

For many reasons, people have designed multiple architectures of data centers such as containerized data centers, industrialized data centers, and traditional data centers. However, the core component in every data center never changed—the primary reason why we built data centers is always to retain servers. Despite the complexity of supporting infrastructures and facilities like Power Distribution Units (PDUs), cooling systems and lightings, data centers are just like shells that provide space for racking physical servers in rows and all necessary conditions to maintain them running 365 × 24 all year around. Playing the most important role in data centers, servers also account for the majority of data center energy consumption—at around 40% of the total and even higher in well-planned data centers with excellent natural/artificial cooling conditions. A single rack of servers can consume more than 20 kW, which is equal to the average power of 35 common households in Austria. In addition to that, the reported power density of server racks is still growing as engineers continue to compact the space but add more servers into modern data centers. All these facts indicate that improving server energy efficiency is a top-priority task when it comes to the effort on energy conservation (and cutting electricity bills, of course) for a data center that has already been operating in the first place.

Energy efficiency can hardly be achieved unless we are able to be precisely aware of how much energy has been used and how quickly our servers are consuming electricity. This means that appropriate setup of power/energy consumption monitoring modules serves as the prerequisites of data center sustainability. Nevertheless, more focus should be cast on how to implement a flexible, fine-grained monitoring system that is easy to scale up with the data center. On this point, we argue that software-based implementation is the right path considering all its advantages (e.g., low cost, scalability and compatibility) match what we need for a decent data center power monitoring system. So, the primary purpose of this chapter is to provide useful guidance and as much information as it is concerned about modeling and then reducing power usage at the server

Data Center Handbook: *Plan, Design, Build, and Operations of a Smart Data Center,* Second Edition. Edited by Hwaiyu Geng.

level—we believe it is a fine granularity considering the massive system of a data center. In the following content, we begin with introducing the common methodologies for server power modeling in the form of taxonomy, including some mathematical stuff by formulating a number of representative power consumption models that are prevailingly used in engineering. Besides, we will look into the problem using some more comprehensive metrics rather than joules and watts by introducing the notion of energy efficiency and discuss the ways to define it. The last part of this chapter is a bit algorithmic as it is all about the topic of energy conservation technology and strategies involving optimization policies from practice and cutting-edge solutions from research, which we believe are of great practical use in establishing a green data center.

19.2 ENERGY CONSUMPTION MODELING OF SERVERS IN DATA CENTERS

19.2.1 Energy Efficiency of Servers in Data Centers

There have been broad concerns about the rapid growth of data center energy consumption as well as their low energy efficiency. The electricity consumption of the ICT sector (within which data centers are the most important operation in industrialized countries) accounts for 5–10% of the total. Improving energy efficiency of data center is an important subject on the way to achieving Green IT, which typically means less greenhouse gas emissions, less harmful material, and encouraging the use of renewable energy.

Using the most intuitive definition, energy efficiency can be measured as the ratio of the amount of work done to the amount of energy consumption over a period of time. The metric is crucial in case you want to look into how efficient and productive your servers are and before you decide to optimize them. Fortunately, there are quite a few energy efficiency metrics available including the commonly used power usage effectiveness (PUE). Nevertheless, some of them are either too simple to reflect the real situations of a server or too complicated to be applied in practical use. So, we believe it is very necessary to single out some metrics concerning server energy efficiency that could be useful in a typical data center.

In practice, we often use floating-point operations per second (FLOPS)/Watt to measure a server's efficiency in energy by putting its performance over its power consumption. The metric is simple but quite useful in showing dynamics of a server provided that it operates at constantly changing levels of workload.

$$\text{Server\_power\_efficiency} = \frac{\text{performance}}{\text{power consumption}}$$

In case metering devices are not always available (which is realistic as attaching every server with a meter is prohibitive), both the performance and the power consumption can be modeled as quadratic functions by fitting data composed of server power consumption and performance [1]. Therefore, the formula of server power efficiency metric can be further reformulated as the following:

$$\text{Server\_power\_efficiency} = \frac{c_0 + c_1 * \mathrm{u} + c_2 * \mathrm{u}^2}{d_0 + d_1 * \mathrm{u} + d_2 * \mathrm{u}^2}$$

$$\mathrm{u} = \text{CPU\_utilization}$$

In the formula, the parameters c_0, c_1, c_2, d_0, d_1, and d_2 are obtained by data fitting based on the data collected from history run traces where an adequate number of power and performance samples are recorded.

We can also exploit the utilization of various resources combined with the server energy efficiency to provide a comparatively more accurate prospective into our monitoring system. One of the widely used representatives is the server Energy-Efficient Utilization Indicator (EEUI) [2], which is defined as follows:

$$\text{Server}_{\text{EEUI}} = \text{Server}_{\text{EEUI}}^{\text{CPU}} + \text{Server}_{\text{EEUI}}^{\text{Memory}} + \text{Server}_{\text{EEUI}}^{\text{Network}} + \text{Server}_{\text{EEUI}}^{\text{Disk}}$$

where $\text{Server}_{\text{EEUI}}^{\text{X}}$ denotes the EEUI of component $X \in \{\text{CPU, Mem, Net, Disk}\}$. Typically, we calculate the EEUI of CPU in the following way:

$$\text{Server}_{\text{EEUI}}^{\text{CPU}} = \frac{\text{EE}_{\text{CPU}}}{\text{EE}_{\text{CPU\_max}}} \times \frac{P_{\text{CPU}}}{P}$$

where EE_{CPU} and $\text{EE}_{\text{CPU\_max}}$ stand for the associated energy efficiency level of CPU mapped from CPU utilization and the maximum energy efficiency that CPU can operate, respectively. P_{CPU} and P denote the power consumed by the CPU and the server-wise total consumption, respectively. For memory, network component, and disk, we use similar formulations to determine their EEUI, and thus we get the server's EEUI as a sum of them. It is worth noting that $\text{Server}_{\text{EEUI}}^{\text{X}}$ is not designed for comparing the energy efficiency between different server hardware or architectures. Instead, it is used to monitor the level of energy efficiency when the servers are operating. In summary, EEUI not only considers the utilization levels at which those components are used but also takes into account their energy efficiency and energy proportionality.

19.2.2 Modeling Methods of Servers' Energy Consumption

With rapid development of hardware, new servers from different vendors have supported accessing their real-time power consumption, but solely using physical measurement is never the most compatible and scalable solution, nor can it predict the future energy demand of the servers. Therefore, establishing an accurate and generic energy consumption model becomes a cornerstone in realizing a mega-scale, real-time, low-cost monitoring system for the sake of optimizing the energy efficiency of the cloud servers.

Based on the methodologies of modeling, we classify the energy consumption models of cloud servers into two categories, namely Performance-Monitor-Counter-based models and resource-utilization-based models. We also introduce the general procedure of modeling briefly.

19.2.2.1 Performance-Monitor-Counter (PMC)-Based Models

Intel has introduced a set of model-specific registers (MSRs) into their performance monitoring systems of processors since Pentium series. The so-called PMC-based model refers to establishing the energy consumption estimation model by analyzing the relationship between server energy consumption and the performance events provided by hardware.

Constructing a PMC-based model consists of three steps:

1. Keep listening for PMC events from subcomponents of servers (e.g., CPU, memory, disk, and Network Interface Controller);
2. Analyze the relationship between specific performance counters and system energy consumption;
3. Establish the relationship model between PMC events and system energy consumption.

The PMC-based model warrants a high accuracy in general, but it may take lots of effort to study which events should be selected because too many events involved could increase the complexity and the risk of over-fitting as well. A defect of PMC-based models is that the model may be invalid as the hardware architecture changes.

19.2.2.2 Resource-Utilization-Based Models

The principle of resource-utilization-based model is to find the correlation between power/energy and resource utilization, which are coarse-grained but easily accessible at the OS level. The measurement of resource utilization can be easily established using existing OS monitoring tools. The most classical form is linear regression model with CPU utilization as the only variable and two parameters of which C_0 is a constant and C_1 is a factor:

$$P = C_0 + C_1 \times u_{\mathrm{CPU}}$$

This type of model is flexible and reliable under a single type of load. However, it may be poor in fitting the power consumption curve when the types of tasks are various and the workload fluctuates frequently and significantly.

19.2.2.3 General Modeling Process of Server Energy Consumption

Generally, the modeling process can be summarized as the following steps (shown as Fig. 19.1): data sampling, data processing, model building and model evaluation.

- Data Sampling
  Data sampling takes two parts of work simultaneously: power consumption sampling and system performance sampling. As Figure 19.2 shows, built-in sensors (e.g., IPMI and RAPL) or external devices can acquire the server power consumption. Monitor tools, like Perf and OProfile, can acquire the system performance data.
- Data Processing
  Missing value processing, denoising techniques, and data normalization are commonly used to preprocess the raw data and analyze the characteristic and potential relationship between the performance and energy consumption.
- Model Building
  Regression methods (e.g., linear/nonlinear regression) or more complex methods (e.g., SVR and neural network) can be used to model the relationship between the input features and server power consumption. After having the basic form of the model, parameters optimization and error correction should be considered to ensure the accuracy.
- Model Evaluation
  After producing a trained model, we should evaluate its accuracy, overhead and other indicators. The model should be tested in the production environment or use a set of benchmarks to simulate the specific task scenario to evaluate the model. Some metrics are widely used to improve the model, such as MSE (mean squared error) and MAPE (mean absolute percentage error).

19.2.3 Power Models of Servers in Data Centers

Mathematically, a power model can be defined as a function that maps the variables related to the system state to the system power consumption, which takes one or more system indicators (e.g., CPU, memory, network adapter and disk

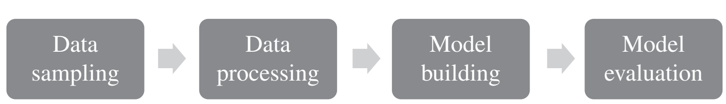

FIGURE 19.1 The workflow of cloud server energy consumption modeling. Source: © 2020 Weiwei Lin.

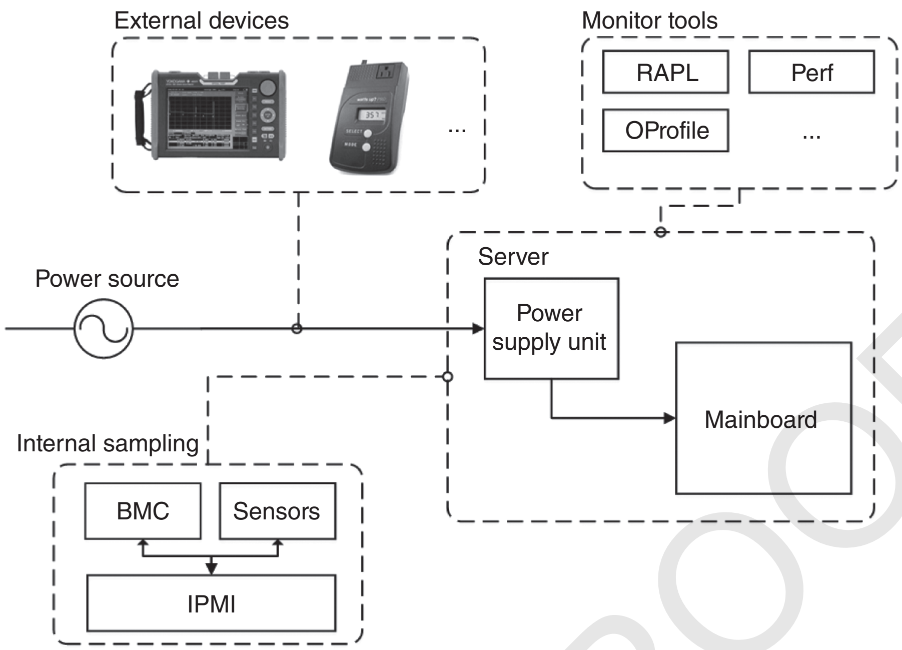

FIGURE 19.2 The data sampling framework of server energy consumption modeling. Source: © 2020 Weiwei Lin.

utilization) as the function's independent variables, and takes the instantaneous power or the cumulative energy within a period of time as the function output. According to the types of server instances, power models of servers can be roughly divided into three categories: power models of the physical server, virtual machine (VM) power models, and container power models (as shown in Fig. 19.3).

No matter what process management or virtualization technology is used, power consumption is ultimately reflected in the fluctuation of workload. At the physical server level, we summarize existing power models into two categories: coarse-grained power models and fine-grained power models. For coarse-grained models, the essence is to screen the underlying complexity and hierarchical structure of physical server and to model the power consumption at the highest level. Namely, it mainly focuses on the power-consuming entities that can perform work independently (such as running operations and cooling). For example, Fan

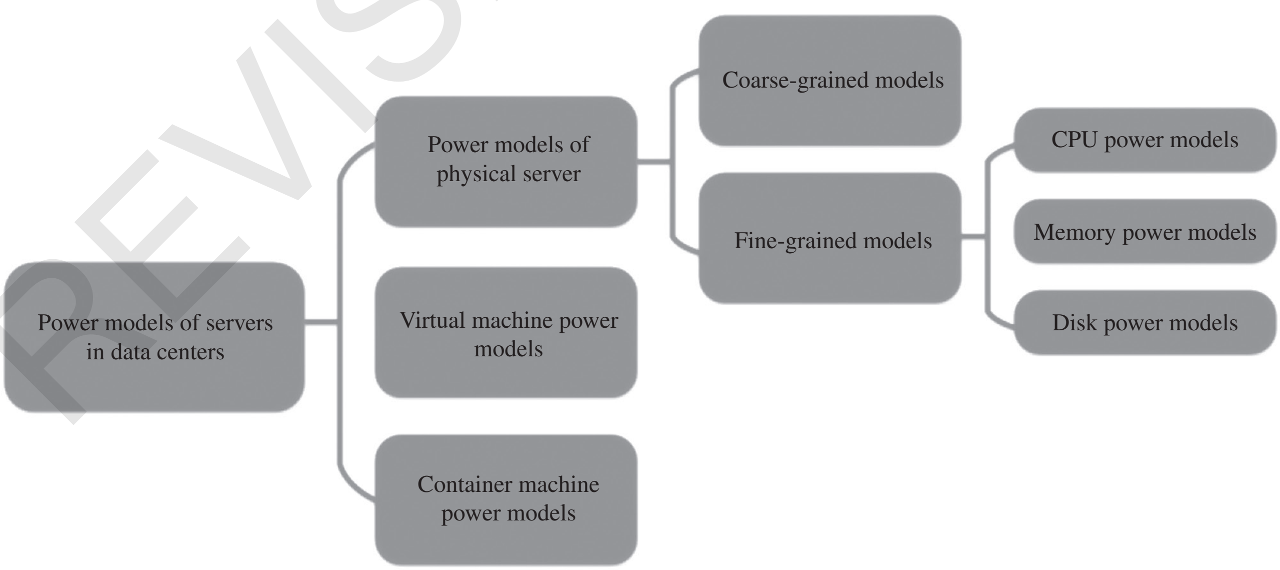

FIGURE 19.3 The power models' specific classification of servers in data center. Source: © 2020 Weiwei Lin.

et al. [3] proposed a power model for estimating the whole physical server power based on CPU utilization:

$$P_{\text{server}}(t) = c_0 + c_1 u(t)$$

where both c_0 and c_1 are model parameters, $u(t)$ represents the CPU utilization. For fine-grained models, we need to consider the energy-consuming entities that are not working independently and model each of them separately. Using linear forms again, we can formulate the power of major components such as CPU [4], memory [5], and disk [6] as below:

$$P_{\text{cpu}} = P_{\text{cpu}_{\text{idle}}} + \left(P_{\text{cpu}_{\text{max}}} - P_{\text{cpu}_{\text{idle}}}\right) u_{\text{cpu}}$$

$$P_{\text{mem}} = P_{\text{mem}_{\text{idle}}} + C \cdot u_{\text{mem}}$$

$$P_{\text{disk}} = P_{\text{disk\_idle}} + C_r m_{\text{read}} + C_w m_{\text{write}}$$

For VM power models, since the management of virtual resources and the physical resources are separate from each other in the virtualization layer, we cannot directly apply the power modeling methods for the traditional hardware layer to VMs. In current researches on VM energy consumption, there are two main methods applicable to modeling VM's power, namely white box methods [7] and black box methods [6]. The main difference is that the former is to embed the monitoring agent into VM and obtain information from the internal, while the latter is to obtain information from the host for modeling and monitoring.

For container power models, since container encapsulates processes and packages and all the resources required for the software to run in an isolated container, and the application process runs directly sharing the host's kernel, it is less intuitive to model the containers' power. Currently, there are only a few studies working on this topic. As a possible solution, some existing work propose to model containers' power at process-level and include machine learning techniques. For example, David et al. [8] introduced a process-level power model, in which the container is treated as a process on its host (a VM or a server). Kang et al. [9] proposed a container power model based on k-medoid clustering, which makes use of the characteristics of both server and container as features.

19.3 ENERGY-SAVING TECHNOLOGIES OF SERVERS

A wide range of technologies and techniques are ready for use if you believe the servers are not running in the optimal style and decided to make some changes. The primary purpose of this part is to provide information and some guidance in terms of how to utilize existing technologies for improving server-wise energy efficiency. Some of the presented ones are cutting-edge from the latest researches and some have been widely adopted in the industry like a rule of thumb or best practice. Anyway, these methods, schemes, and techniques to be introduced have more or less proved to be effective or at least helpful when you are looking for solutions that can make your servers run in a more power-efficient manner.

It is a complex problem to characterize what a server is doing with every joule of energy given to it. The power models detailed in the previous sections could be helpful for understanding each factor that contributes to the total consumption of a server when we break it down into components. From that some may think that energy could be saved as long as we remove some of the components, and the industry knows it well—"You had the opportunity to strip things down to just what you need, and make it specific to your application," says Bill Carter, chief technical officer at the Open Compute Project. However, people do not truly lose weight by not wearing clothes, so do the servers. The most effective way to make servers less hungry to energy is to manage them wisely—which is exactly the topic of this section. With the advances of both research and practice, existing energy-saving techniques for servers have covered a broad range of applications including hardware-based, software-based, and more. Actually, some of the giant companies, like Facebook, have deployed them comprehensively in their data centers around the globe. To make it clear and organized, these techniques will be introduced as they are summarized in the following categories:

- Dynamic Server Management: includes a handful of both relatively conventional power-saving techniques that may need certain hardware function and state-of-the-art algorithms that make use of artificial intelligence (AI) to manage servers in an automatic fashion.
- Task Scheduling: about software-based high-level optimization of server workload by rearranging a group of tasks in given workflows for achieving shorter makespan and less energy consumption.
- VM Allocation and Consolidation: implementation of VM management algorithms that produce the best match between physical servers and VMs and energy-optimal VM migration plans.
- Light-weight Virtualization: a bunch of emerging implementations of resource encapsulation and isolation as lower-cost virtualization technology.
- Load Scheduling with Renewable Energy Provisioning: incorporates scheduling algorithms and frameworks that leverage the dynamics of renewable energy sources to reduce the expensive grid power usage.

19.3.1 Dynamic Server Management

In 2017, Jonathan Koomey, a California-based consultant and leading international expert on IT, surveyed with a colleague more than 16,000 commercial servers and found that about one-quarter of them were "zombies," gobbling up power without being useful. As a matter of fact, zombie servers may not be rare in large-scale data centers but by many means these power eaters can be avoided.

19.3.1.1 Dynamic Scale-Out/Scale-In

Managing servers is kind of like managing your employees—a popular management technique is to make sure that every one of them runs at full throttle as much of the time as possible, whereas others are turned off/sacked rather than being left idle. Facebook invented a system called *Autoscale* [10] that, as they claim, can reduce the number of servers that need to be on during low-traffic hours, and this led to power savings of about 10–15% in trials.

They implemented a specifically designed load-balancing framework that aggregates the workload on a subset of their entire cluster by rerouting the coming requests. By doing so, there is a big chance that a portion of their servers in the data center will have nothing to do if it is not in peak hours. Once a server turns idle for a certain length of period, it will be shut down or switch to some energy-saving mode—in either way it goes inactive and consumes much less energy. The strategy usually works as the coming workload generally follows some time series patterns that have a lot to do with the habits of users. By setting up rules that prioritizes some of your servers, it is not difficult to get only a group of them to work rather than spreading the load across your cluster stochastically. Nevertheless, the main challenges are how to accurately anticipate workload (e.g., the number of coming requests) so that you don't activate and deactivate servers frequently and how to achieve real-time elasticity by making use of your anticipations.

19.3.1.2 Dynamic Work Mode Switching

Frequently powering on/off servers may cause prohibitive overheads in both energy cost (we often see power peaks in the start-up process) and server lifespan. In terms of energy savings, Facebook manages to scale in their cluster by deactivating servers in a much softer manner - putting inactive servers into power-saving mode. Averagely, with the technique introduced, they are able to achieve >10% power saving over a 24-hour cycle for different web clusters.

It is worth mentioning that mode switching and power-saving tweaks have been well supported by multiple mainstream operating systems including Windows server (supporting six levels of power-saving states from S0 to S5) and many Linux distributions (e.g., the *pm-utils* and component-wise power-saving tweaks for Ubuntu and the *CPUfreq governor* for Red Hat)**.**

19.3.1.3 DVFS and Alternatives

Server power usage has a strong correlation with CPU utilization, making dynamic tuning of CPU state a vital part of server power management. Dynamic Voltage Frequency Scaling (DVFS) is one of the most notable techniques for reducing CPU power consumption and it works based on the following CMOS power principles:

$$P = \alpha C V^2 F$$

where αC is a constant for a specific processor, V and F denote the supply voltage and work frequency, respectively. P is the instant power consumption of CPU, which we want to reduce in case there is not much workload on the server. From the equation, it is pretty clear that DVFS techniques (along with associated techniques such as dynamic voltage scaling (DVS) and adaptive voltage and frequency scaling (AVFS)) are very effective in energy conservation, since lowering the voltage (V) has a squared effect on active power consumption while performance degradation (dependent on F) is basically linear. Specifically, some experiments indicate that with DVFS it is possible to achieve a 3× reduction in power with only 1× reduction in performance.

As an alternative, AVFS is an extension of DVFS. DVFS is usually limited to scaling in a series of fixed discrete steps in terms of the voltage and frequency of the targeted power domains, making it an open-loop system with large margins built in, and therefore the power reduction is not optimal. On the other hand, AVFS deploys closed-loop voltage scaling and is compensated for variations in temperature, process, and IR drop via dedicated circuitry that constantly monitors performance and provides active feedback. Although the control is more complex, the payoff in terms of power reduction is higher.

An obvious side-effect of scaling down CPU voltage/frequency is that it takes you a longer time to finish a task compared to that needed in the max-performance mode, and that in turn lessens the benefit you gain from the reduction of power (see the following equation).

$$E = P \cdot t = \alpha C V^2 F \cdot t$$

Generally speaking, voltage and frequency scaling techniques are very useful when applied together with server mode switching to minimize energy wastage in underutilized or idle states, but its benefits could be marginal in case of continuous peak loads.

19.3.1.4 Proactive Load Control

As effective as DVFS and AVFS are, they are designed to rein in the power of a server in a totally reactive manner. However, when your servers are overwhelmed by workload (e.g., bursts of incoming requests on Black Friday), hardware-based scaling may only make little difference and even lead to system instability.

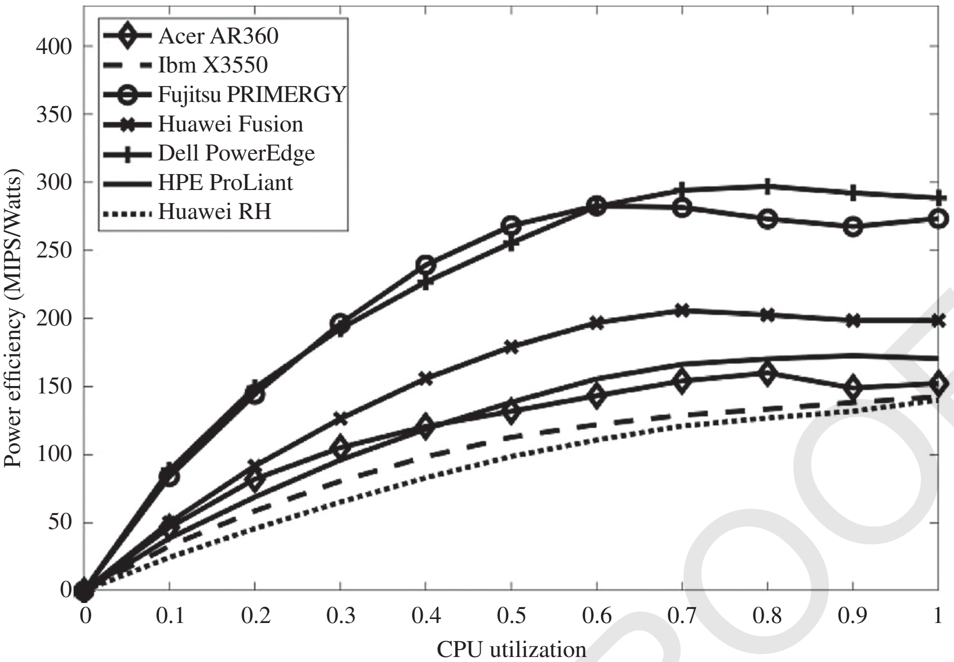

FIGURE 19.4 Power efficiency of seven different servers computed according to the data provided by SPEC [11]. Source: Lin et al. [12]. © IEEE.

It is always important to keep the load in check no matter from the perspective of service quality or the standpoint of energy conservation. For one thing, the risk that your server gets burned out certainly increases when it is working under constant high load (e.g., CPU usage >95%). For another, research results (see Fig. 19.4) have shown that keeping servers running at their maximum utilization is often not energy efficient—a large number of them attain the optimal performance–power ratio at a load level around 80%.

Proactive load control is very necessary so that we can have enough time and resources to make use of the power characteristics of physical servers. In fact, many studies (e.g., [13, 14]) have suggested managing the load of servers proactively and preventively. This, in many cases, requires the system to be always aware of the workload level on each server, and to redirect or deny some requests once it shows signs of being overloaded. Proactive load control helps to make sure the operators have time to scale up and prevent the jittering workload from affecting server power efficiency.

19.3.1.5 Comprehensive Optimization with AI

In the past few years, empirical analysis and expertise are literally the rules of thumb when it comes to find the optimal configuration and operational policies for a system—server management in data centers was of no exception. But that seems to change recently as the rapid advance of AI has drawn world-wide attention, and AI-driven solutions did come out with a lot of success in a variety of domains that people reckon as collections of complex problems for humans.

As it has been mentioned, dynamic power management of servers is intricate because no one tells you how to find out the optimal configuration of DVFS or what turns out to be the best policy for load management. So the question is: can AI be the leading light of energy optimization of servers? Most people would say yes and maybe more when they see the promising results from some state-of-the-art studies. For example, the authors of paper *Automated cloud provisioning on AWS using deep reinforcement learning* [15] innovatively propose to adopt Deep Reinforcement Learning (RL) to realize automatic cluster scale-out/in. They built a smart cluster controller based on Q-learning—a popular RL method—by modeling *Q-state* as the number of server instances, *action* as the decision of scaling-out or scaling-in, and *reward* as the resulting change of energy consumption (Q-learning, in essence, learns how to take good actions in a given Q-state considering the resulting state and reward). Figure 19.5 shows the typical architecture of using (deep) Q-learning network for automated resource provisioning on cloud infrastructures where the "Environment" represents a circumstance that continuously provides feedbacks ("state change" and "reward") to the model.

After certain rounds of training, the RL-driven controller shows its capability in finding energy-efficient decisions and it is further claimed that it is also able to learn policies that balance performance and energy cost—as long as humans specify what they want. Something even more amazing is

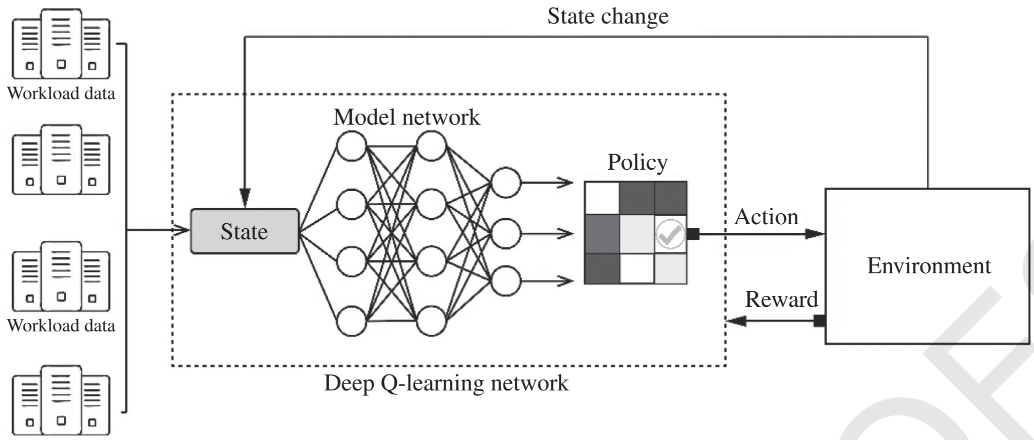

FIGURE 19.5 A schematic diagram showing a general architecture of (deep) Q-learning network where the specific structure of the underlying model network can be fully connected, convolutional, or a combination of both. Source: © 2020 Weiwei Lin and Wentai Wu.

that the learned policies are refined over time because RL enables the controller to constantly improve as it works. This is only a small case where one can apply the spell of AI to server power management in modern data centers. Actually, AI and machine learning can do more, and these emerging techniques are promising in taking the place of human experience (at lease, partially) in terms of comprehensive energy usage optimization.

19.3.2 Task Scheduling

Task scheduling is one of the most fundamental problems when it comes to optimizing resource provisioning on servers given a flow of tasks (which in some cases are considered as decomposed user jobs). In general, task schedulers and associated job coordinators are implemented at the software level. Depending on the optimization target, they can have major impact on the servers' productivity and, of course, energy consumption.

Given a batch of tasks, a task scheduler basically needs to decide which task should be assigned to which server and what order should the co-allocated tasks be executed. If you are familiar with the classical bin-packing problem, you can easily find the similarity as well as the complexity in solving it—it is a combinatorial NP-hard problem. Because of this, heuristic approaches (e.g., greedy algorithms and evolutionary algorithms [16]) are commonly used for the problem of task scheduling. For example, *Min-Min* [17] is one of the most famous solutions and it proved its effectiveness in shortening average task makespan (which thereby helps reduce energy usage). However, in cloud data centers, which are usually hyper-scale and virtualized, tasks are most often bound to virtual machines (VMs) to realize resource isolation. The additional role of VMs increases the complexity of task scheduling but many studies have already come up with promising solutions. Lin and Wu [18] introduced an energy-aware task-to-VM scheduling framework (Fig. 19.6) that generates energy-efficient allocation plans by comprehensively considering task demands,

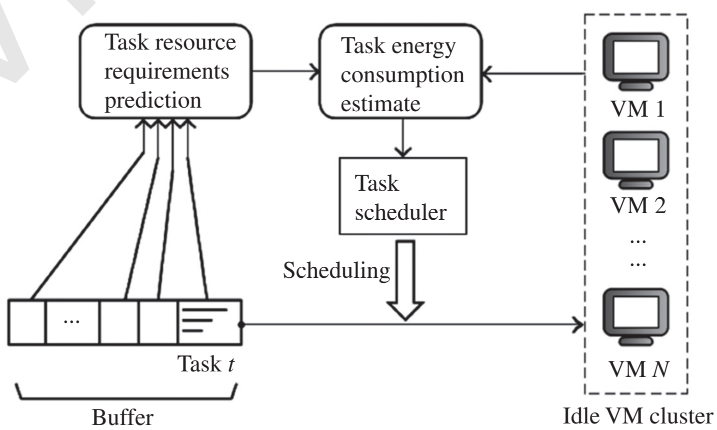

FIGURE 19.6 An overview of an energy-aware scheduling framework that caches arrived tasks in a buffer, estimate task energy consumption, and invoke scheduling functions to allocate tasks to VM instances running on active servers. Source: © 2020 Weiwei Lin and Wentai Wu.

VM's power efficiency as well as server workload. Their algorithm achieves a reduction of over 20% in energy consumption across the cluster.

19.3.3 VM Allocation and Consolidation

Virtualization has been prevailingly adopted in data centers since the last decade, and there are a lot of research and practice focusing on how to maximize the benefits we gain from virtualization through VM orchestration. It is not a one-shot operation as we need to continuously handle newly created instances as well as readjust the location of running VMs for the purposes of both load balancing and energy conservation.

VM allocation refers to a range of strategies that optimize the mapping from VMs to bare-metals and VM consolidation mainly deals with the reallocation of VMs by means of migration. In order to increase server utilization and reduce energy consumption, cloud infrastructure providers pay very much attention to dynamic VM allocation/reallocation since emerging techniques have made live migration operations faster than ever. Compared to allocation, reallocation of VMs requires the algorithms to be a bit more sophisticated—it is a multistep operation that starts from detecting overloaded/underutilized hosts (i.e., physical servers) first, then picking one or more VMs from these hosts for migration, and finally finding a group of candidate servers that are competent as target hosts. Each step could be intricate as a lot of energy-related information and consideration are needed to make the overall process worthwhile with respect to energy reduction. Figure 19.7 displays a schematic framework showing the basic workflow and cooperation between modules that support energy-saving VM allocation and consolidation.

As complex as the framework is, it is always important to keep in mind that the overheads of VM allocation/reallocation must be strictly in check. This may unfortunately restrain you from applying some novel but high time-complexity algorithms into your VM management strategy. Empirically speaking, there is a trade-off between the optimality (i.e., how much energy you can save) and the efficiency (i.e., how long it takes to make a decision) of your strategy, and this should be well considered at the very beginning.

19.3.4 Light-Weight Virtualization

Engineers are always looking for something more efficient and cost-saving and they began to lose their interests in traditional virtualization technology as VMs are so "clumsy" if you look at how many resources are needed to run a hypervisor on your server. Since VMs are too heavy, light-weight virtualization technology earned popularity quickly and the most prevailing branch today is the container (or containerization).

Having its name from the shipping industry, container technology refers to a method for packaging an application so it can be run, with its dependencies, isolated from other processes. The mainstream public cloud computing providers, including Amazon Web Services (AWS), Microsoft

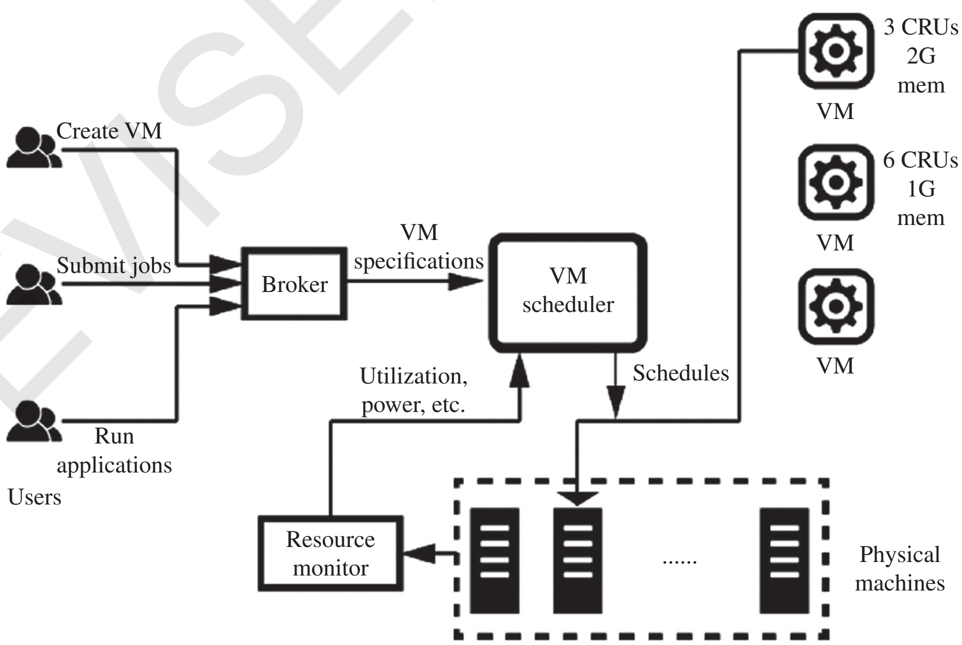

FIGURE 19.7 A virtual machine (re)allocation framework for clouds providing flexible services adopted in the study on energy-efficient cluster management. Source: Lin et al. [12]. © IEEE.

Azure, and Google Cloud Platform, have embraced container technology and widely deployed it in their hyperscale data centers. The main strength of containers, compared to VMs, is that they share the OS kernel and do not require the overhead of associating an operating system within each application. Because of this, containers are far smaller in capacity than a VM and require less start-up time, allowing a lot more instances running on the same server. This drives higher server efficiencies and, in turn, reduces energy cost for running a same number of applications. To date, there are more than a handful of options if one decides to put containerization into practice, here are some of the most popular implementations: *Linux LXC*, *Docker, KataContainer* (by OpenStack), *Shifter* (by IBM), and *FireCracker* (by Amazon).

Another trending technology that shows it promise in further amplifying the benefits of light-weight virtualization is called Unikernel. Also known as container 2.0, the design principle behind Unikernel is a further step into minimalism—provide just enough software to power the desired application and nothing more. Technically speaking, Unikernels rely on specialized compilers to combine application software and supporting OS functions at compile time instead of runtime. This results in a single application image that contains everything the application needs to run. All drivers, all I/O routines, and all supporting library functions normally provided by an operating system are included in the executable, while those unneeded ones will be excluded in the image to keep it as light as possible. For example, *MirageOS* (an established Unikernel project) claimed that they have a working domain name server that compiles into just 449 kB. The project also has a web server that weighs at 674 kB and an *OpenFlow* learning switch that tips the scales at just 393 kB.

The development of virtualization is heading to the direction where resource-efficiency is put on the first place. This is a good news from the perspective of energy conservation—servers theoretically get more power efficient when we keep reducing the overhead of virtualization. But extra benefits always come at a cost, and the cost here is weakened isolation between applications, which may raise some issues concerning both security and resource contention.

19.3.5 Load Scheduling with Renewable Energy Provisioning

In 2011, Facebook made a commitment to using 100% renewable energy. Google (the largest corporate purchaser of renewable energy on the planet so far) and Apple followed in 2012. As of 2017, nearly 20 Internet companies had done the same. Electricity from renewable energy sources was not so advocated by the ICT industry at the beginning of this decade—back in 2010, IT companies were

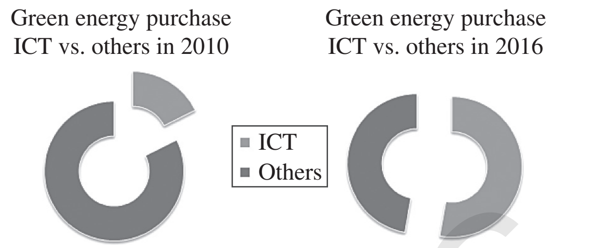

FIGURE 19.8 The contribution to renewable electricity contracts by the ICT industry ramped up between 2010 and 2016, according to the report by International Energy Agency (IEA) [20]. Source: © 2020 Weiwei Lin and Wentai Wu.

still a negligible contributor to renewable-power purchase agreements with energy suppliers; but by 2015, they accounted for more than half of such agreements [19] (see Fig. 19.8).

The reason why ICT giants show tremendous interest in introducing renewable power provisioning to their data centers are multifold. First, the government keeps forcing enterprises to reduce their carbon footprint, and apparently replacing brown energy sources with green ones is one of the best options. Besides, the prices of renewable electricity are expected to decrease as people believe it is very hopeful to embrace new technologies that make renewable power generation (e.g., wind and solar power) more efficient than ever.

Data center owners can benefit more from what renewable energy sources already brought to us. Related studies have shown the great potential in further reducing server energy cost through the refinement on workload management by leveraging the characteristics of green power supply. For example, researchers integrated a renewable energy supply prediction model into task scheduling algorithms to rearrange the execution order of tasks so as to maximize the utilization of renewable electricity [21] (see Fig. 19.9 where the "green energy" is used whenever available, while electricity supply from traditional power grid accounts for the rest), and a novel technique called Battery Assisted Green Shifting is introduced to increase the flexibility in the way renewable energy is used for the execution of jobs on a server.

These cutting-edge studies provide very useful insights into what we can do in the operation of data centers when we have the chance to power our servers with renewable energy. We can foresee that there will be more opportunities as well as challenges in the (near) future. With the evolutionary force of green power, maybe we can raise the energy efficiency of servers and data centers to a level higher than ever, or maybe data centers can totally get rid of traditional power grids, or at least cloud services could be repriced based on how much green energy is used. Only time will tell.

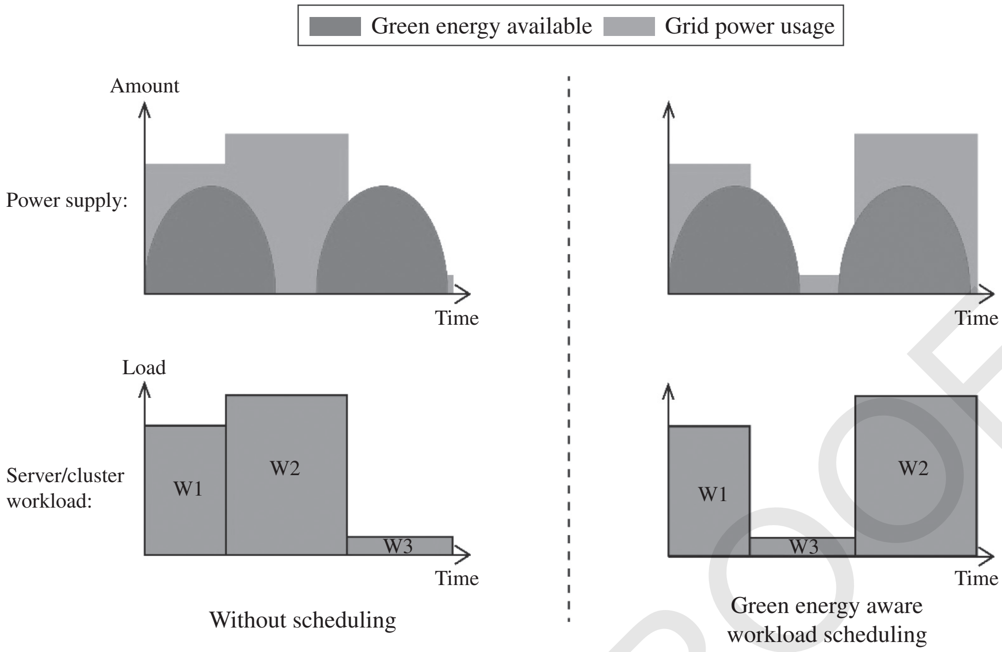

FIGURE 19.9 A schematic showing how to reschedule cloud workload based on the dynamic of power supply to maximize the utilization of green/renewable energy. Source: © 2020 Weiwei Lin and Wentai Wu.

19.4 CONCLUSIONS

Considering the significance of energy consumption, this chapter conducted a systematic, in-depth study about energy consumption and energy-saving technologies of servers in data centers. To better understand, we first introduced the modeling energy consumption of servers by presenting the modeling methods and general modeling process of server energy consumption. Next, the power models of servers in data centers are introduced with a hierarchy, focusing on the power models of the physical server, VM, and container. And then, the energy efficiency of servers in data centers was presented to evaluate the power models of servers, which mainly includes the energy efficiency metrics and the examples of server energy efficiency. Moreover, to better provide information and some guidance on how to utilize existing technologies for improving server-wise energy efficiency, the energy-saving technologies of servers including the dynamic server management, task scheduling, VM allocation and consolidation, light-weight virtualization, and load scheduling with renewable energy provisioning are also presented in details.

Simultaneously, we observed that there have been a large number of studies conducted on the energy consumption and energy-saving technologies at the lower levels of the data center, but much less work has been done at the higher levels. And this is the most prominent problem in the current research on energy consumption modeling and energy-saving technology for data centers. Therefore, we need to further explore higher-level issues (such as orchestration of containers) in future research.

ACKNOWLEDGMENTS

This chapter is partially supported by the National Natural Science Foundation of China (Grant Nos. 62072187, 61872084) and Major Program and of Guangdong Basic and Applied Research (2019B030302002). We would like to thank Fang Shi (SCUT, China), Gangxing Wu (SCUT, China), Tianhao Yu (SCUT, China), Gaofeng Peng (SCUT, China), Chennian Xiong (SCUT, China), and Hongping Zhan (SCUT, China) for their editing and data collection of this chapter.

REFERENCES

[1] Lin W, et al. A heuristic task scheduling algorithm based on server power efficiency model in cloud environments. Sustain Comput Inf Syst 2018;20:56–65.

[2] Abaunza F, Hameri AP, Niemi T. EEUI: a new measure to monitor and manage energy efficiency in data centers. Int J Product Perform Manag 2018;67(1):111–127.

[3] Fan X, Weber WD, Barroso LA. Power provisioning for a warehouse-sized computer. ACM SIGARCH Comput Archit News 2007;35(2). ACM.

[4] Hsu CH, Poole SW. Power signature analysis of the SPECpower_ssj2008 benchmark. (IEEE ISPASS) IEEE International Symposium on Performance Analysis of Systems and Software; April 10–12; Austin, TX, USA 2011. IEEE.

[5] Lin W, et al. A cloud server energy consumption measurement system for heterogeneous cloud environments. Inf Sci 2018;468:47–62.

[6] Kansal A, et al. Virtual machine power metering and provisioning. Proceedings of the 1st ACM symposium on Cloud computing; June 10–11; Indianapolis, Indiana, USA 2010. ACM.

[7] Li Y, et al. An online power metering model for cloud environment. 2012 IEEE 11th International Symposium on Network Computing and Applications; August 23–25; Cambridge, MA, USA 2012. IEEE.

[8] Snowdon DC, et al. A platform for os-level power management. The European Professional Society on Computer Systems 2009; April 1–3; Nuremberg, Germany 2009.

[9] Kang DK, et al. Workload-aware resource management for energy efficient heterogeneous docker containers. 2016 IEEE Region 10 Conference (TENCON); November 22–25;Singapore, Singapore 2016. IEEE.

[10] Facebook Engineering. Autoscale. Available at https://engineering.fb.com/production-engineering/making-facebook-s-software-infrastructure-more-energy-efficient-with-autoscale/.

[11] SPEC. Available at http://www.spec.org/power_ssj2008/results/.

[12] Lin W, Wu W, He L, Li K. An on-line virtual machine consolidation strategy for dual improvement in performance and energy conservation of server clusters in cloud data centers. IEEE Trans Serv Comput 2019;2019, Doi: https://doi.org/10.1109/TSC.2019.2961082.

[13] Adams E. Optimizing preventive service of the software products. IBM J Res Dev 1984;28(1):2–14.

[14] Castelli V, Harper RE, Heidelberger P, Hunter SW, Trivedi KS, Vaidyanathan KV, Zeggert WP. Proactive management of software aging. IBM J Res Dev 2001;45(2):311–332.

[15] Wang Z, Gwon C, Oates T, Iezzi A. 2017. Automated cloud provisioning on AWS using deep reinforcement learning. arXiv preprint arXiv:1709.04305.

[16] Pacini E, Mateos C, Garino CG. Balancing throughput and response time in online scientific clouds via ant colony optimization (sp2013/2013/00006). Adv Eng Softw 2015;84:31–47.

[17] Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund RF. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. J Parallel Distrib Comput 1999;59(2):107–131.

[18] Lin W, Wu W, Wang JZ. A heuristic task scheduling algorithm for heterogeneous virtual clusters. Sci Program 2016;2016.

[19] Goiri Í, Beauchea R, Le K, Nguyen TD, Haque ME, Guitart J, Torres J, Bianchini R. Greenslot: scheduling energy consumption in green datacenters. SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis; November 12–18; Seattle, Washington, USA 2011, pp. 1–11. IEEE.

[20] International Energy Agency (IEA). Digitalization and Energy; 2017.

[21] Niu Z, He B, Liu F. Not all joules are equal: towards energy-efficient and green-aware data processing frameworks. 2016 IEEE International Conference on Cloud Engineering (IC2E); April 4–8; Berlin, Germany 2016. pp. 2–11. IEEE.