Data Center Energy Consumption Modeling: A Survey

Miyuru Dayarathna, Yonggang Wen, *Senior Member, IEEE*, and Rui Fan

***Abstract*—Data centers are critical, energy-hungry infrastructures that run large-scale Internet-based services. Energy consumption models are pivotal in designing and optimizing energy-efficient operations to curb excessive energy consumption in data centers. In this paper, we survey the state-of-the-art techniques used for energy consumption modeling and prediction for data centers and their components. We conduct an in-depth study of the existing literature on data center power modeling, covering more than 200 models. We organize these models in a hierarchical structure with two main branches focusing on hardware-centric and software-centric power models. Under hardware-centric approaches we start from the digital circuit level and move on to describe higher-level energy consumption models at the hardware component level, server level, data center level, and finally systems of systems level. Under the software-centric approaches we investigate power models developed for operating systems, virtual machines and software applications. This systematic approach allows us to identify multiple issues prevalent in power modeling of different levels of data center systems, including: i) few modeling efforts targeted at power consumption of the entire data center ii) many state-of-the-art power models are based on a few CPU or server metrics, and iii) the effectiveness and accuracy of these power models remain open questions. Based on these observations, we conclude the survey by describing key challenges for future research on constructing effective and accurate data center power models.**

***Index Terms*—Data center, energy consumption modeling, energy efficiency, cloud computing.**

I. Introduction

DATA centers are large scale, mission-critical computing infrastructures that are operating around the clock [1], [2] to propel the fast growth of IT industry and transform the economy at large. The criticality of data centers have been fueled mainly by two phenomenons. First, the ever increasing growth in the demand for data computing, processing and storage by a variety of large scale cloud services, such as Google and Facebook, by telecommunication operators such as British Telecom [3], by banks and others, resulted in the proliferation of large data centers with thousands of servers (sometimes with millions of servers). Second, the requirement for supporting a vast variety of applications ranging from those that run for a few seconds to those that run persistently on shared hardware platforms [1] has promoted building large scale computing infrastructures. As a result, data centers have been touted as one of the key enabling technologies for the fast growing IT industry and at the same time, resulting in a global market size of 152 billion US dollars by 2016 [4]. Data centers being large scale computing infrastructures have huge energy budgets, which have given rise to various energy efficiency issues.

Energy efficiency of data centers has attained a key importance in recent years due to its (i) high economic, (ii) environmental, and (iii) performance impact. First, data centers have high economic impact due to multiple reasons. A typical data center may consume as much energy as 25,000 households. Data center spaces may consume up to 100 to 200 times as much electricity as standard office space [5]. Furthermore, the energy costs of powering a typical data center doubles every five years [1]. Therefore, with such steep increase in electricity use and rising electricity costs, power bills have become a significant expense for today's data centers [5], [6]. In some cases power costs may exceed the cost of purchasing hardware [7]. Second, data center energy usage creates a number of environmental problems [8], [9]. For example, in 2005, the total data center power consumption was 1% of the total US power consumption, and created as much emissions as a mid-sized nation like Argentina [10]. In 2010 the global electricity usage by data centers was estimated to be between 1.1% and 1.5% of the total worldwide electricity usage [11], while in the US the data centers consumed 1.7% to 2.2% of all US electrical usage [12]. A recent study done by Van Heddeghem *et al.* [13] has found that data centers worldwide consumed 270 TWh of energy in 2012 and this consumption had a Compound Annual Growth Rate (CAGR) of 4.4% from 2007 to 2012. Due to these reasons data center energy efficiency is now considered chief concern for data center operators, ahead of the traditional considerations of availability and security. Finally, even when running in the idle mode servers consume a significant amount of energy. Large savings can be made by turning off these servers. This and other measures such as workload consolidation need to be taken to reduce data center electricity usage. At the same time, these power saving techniques reduce system performance, pointing to a complex balance between energy savings and high performance.

The energy consumed by a data center can be broadly categorized into two parts [14]: energy use by IT equipment (e.g., servers, networks, storage, etc.) and usage by infrastructure facilities (e.g., cooling and power conditioning systems).

Manuscript received January 15, 2015; revised July 20, 2015; accepted August 26, 2015. Date of publication September 28, 2015; date of current version January 27, 2016. This work was supported in part by the Energy Innovation Research Program (EIRP), administrated by Energy Innovation Programme Office (EIPO), Energy Market Authority, Singapore. This work was also supported in part by gift funds from Microsoft Research Asia and Cisco Systems, Inc.

The authors are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: miyurud@ntu.edu.sg; YGWEN@ntu.edu.sg; FanRui@ntu.edu.sg).

Digital Object Identifier 10.1109/COMST.2015.2481183

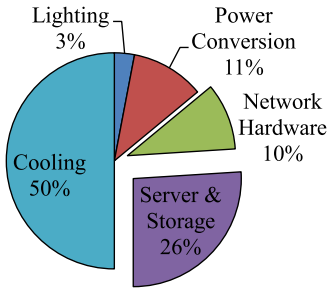


Fig. 1. A breakdown of energy consumption by different components of a data center [15]. The cooling infrastructure consumes a major portion of the data center energy followed by servers and storage, and other infrastructure elements.

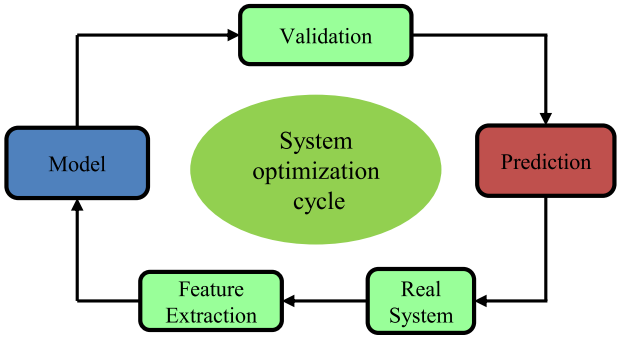


Fig. 2. A systematic view of the energy consumption modeling and prediction process. The data center system optimization cycle consists of four main steps: feature extraction, model construction, model validation, and usage of the model.

The amount of energy consumed by these two subcomponents depend on the design of the data center as well as the efficiency of the equipment. For example, according to the statistics published by the Infotech group (see Fig. 1), the largest energy consumer in a typical data center is the cooling infrastructure (50%) [13], [15], while servers and storage devices (26%) rank second in the energy consumption hierarchy. Note that these values might differ from data center to data center (see for example [16]). In this paper we cover a broad number of different techniques used in the modeling of different energy consuming components.

A general approach to manage data center energy consumption consists of four main steps (see Fig. 2): feature extraction, model construction, model validation, and application of the model to a task such as prediction.

- Feature extraction: In order to reduce the energy consumption of a data center, we first need to measure the energy consumption of its components [17] and identify where most of the energy is spent. This is the task of the feature extraction phase.
- Model construction: Second, the selected input features are used to build an energy consumption model using analysis techniques such as regression, machine learning, etc. One of the key problems we face in this step is that certain important system parameters such as the power consumption of a particular component in a data center cannot be measured directly. Classical analysis methods may not produce accurate results in such situations, and machine learning techniques may work better. The outcome of this step is a power model.
- Model validation: Next, the model needs to be validated for its fitness for its intended purposes.
- Model usage: Finally, the identified model can be used as the basis for predicting the component or system's energy consumption. Such predictions can then be used to improve the energy efficiency of the data center, for example by incorporating the model into techniques such as temperature or energy aware scheduling [18], dynamic voltage frequency scaling (DVFS) [19]–[21], resource virtualization [22], improving the algorithms used by the applications [23], switching to low-power states [24], power capping [25], or even completely shutting down unused servers [10], [26], etc. to make data centers more energy efficient. However, we note that having an energy model is not always necessary for energy consumption prediction.

A *model* is a formal abstraction of a real system. Models for computer systems can be represented as equations, graphical models, rules, decision trees, sets of representative examples, neural networks, etc. The choice of representation affects the accuracy of the models, as well as their interpretability by people [27]. Accurate power consumption models are very important for many energy efficiency schemes employed in computing equipment [28]. Multiple uses for power models exist, including

- Design of data center systems: Power models are necessary in the initial design of components and systems, since it is infeasible to build physical systems to assess every design choice's effect on power consumption [28]. As an example, this approach was used for Data Center Efficiency Building Blocks project by Berge *et al.* [29].
- Forecasting the trends in energy efficiency: In daily operations of computer systems, users and data center operators need to understand the power usage patterns of computer systems in order to maximize their energy efficiency. Physical power measurement alone does not provide a solution since they cannot predict future power consumption, a.k.a. "what if" scenarios [30]. Measurements also do not provide a link between resource usage and power consumption [28]. Experimental verification using real test data is generally expensive and inflexible. Energy models on the other hand are much cheaper and more adaptive to changes in operating parameters [31].
- Energy consumption optimization: Many different power consumption optimization schemes have been developed on top of power consumption models which are represented as mathematical functions [32].

Power modeling is an active area of research, studying both linear and nonlinear correlations between the system utilization and power consumption [33].

However, modeling the exact energy consumption behavior of a data center, either at the whole system level or the individual component level, is not straightforward. In particular, data

center energy consumption patterns depend on multiple factors such as hardware specifications, workload, cooling requirements, types of applications, etc., which cannot be measured easily. The power consumed by hardware, software that runs on hardware, and the cooling and power infrastructure of the building in which the data center systems reside are all closely coupled [34]. Furthermore, it is impractical to perform detailed measurements of the energy consumption of all lower level components, since the measurement infrastructure introduces overhead to the system. Due to these reasons energy consumption prediction techniques have been developed which can estimate the level of energy consumed by a system for a given workload. Energy consumption prediction techniques can also be utilized for forecasting the energy utilization of a given data center operating in a specific context.

The contributions of this paper are numerous. One of the key contributions of this survey is to conduct an in-depth study of the existing work in data center power models, and to organize the models using a coherent layer-wise abstraction as shown in Fig. 4. While there are many current power models for different components of a data center, the models are largely unorganized, and lack an overall framework that allows them to be used together with each other to model more sophisticated and complex systems. Furthermore, we give a more detailed taxonomy of the makeup of a data center, as shown in Fig. 6, and again place and relate existing work to our taxonomy. We believe the breadth and organization of our approach makes this survey a valuable resource for both researchers and practitioners seeking to understand the complexities of data center energy consumption at all levels of the system architecture.

The rest of this paper is organized as shown in the Table I.

TABLE I
CONTENTS

II. RELATED SURVEYS

While there has been a wide body of research on energy consumption modeling and energy consumption prediction for data centers, there has been relatively few surveys conducted in this area. The surveys published till present can be classified under five categories: computing, storage and data management, network, infrastructure, and interdisciplinary.

The majority of existing surveys have been on the energy consumption of computing subsystems. For example, the survey by Beloglazov *et al.* described causes for high power/energy consumption in computer systems and presented a classification of energy-efficient computer designs. However, this survey was not specifically focused on energy consumption modeling. Venkatachalam *et al.* conducted a survey on techniques that reduce the total power consumed by a microprocessor system over time [35]. Mittal's survey on techniques for improving energy efficiency in embedded computing systems [42] is in the same line as Venkatachalam *et al.* work. However, both these works focused on embedded systems, whereas our focus is on data centers, a far different type of system. Mittal *et al.* presented a survey on GPU energy efficiency [43]. Reda *et al.* conducted a survey on power modeling and characterization of computing devices [38]. They reviewed techniques for power modeling and characterization for general-purpose processors, system-on-chip based embedded systems, and field programmable gate arrays. The survey conducted by Valentini *et al.* studied characteristics of two main power management techniques: static power management (SPM) and dynamic power management (DPM) [53].

Several surveys have been conducted focusing on storage and data management in data centers. A survey on energy-efficient data management was conducted by Wang *et al.* [37]. Their focus was on the domain of energy-saving techniques for data management. Similarly, Bostoen *et al.* conducted a survey on power-reduction techniques for data center storage systems [40]. Their survey focused only on the storage and file-system

TABLE II
COMPARISON OF RELATED SURVEYS

Year	Investigator(s)	Area of focus
2005	Venkatachalam *et al.* [35]	Power consumption of microprocessor systems
2011	Beloglazov *et al.* [36]	Energy-efficient design of data centers and cloud computing systems
2011	Wang *et al.* [37]	Energy-saving techniques for data management
2012	Reda *et al.* [38]	Power modeling and characterization for processors
2012	Valentini *et al.* [38]	Power management techniques
2013	Ge *et al.* [39]	Energy efficiency of data centers and content delivery networks
2013	Bostoen *et al.* [40]	Power-reduction techniques for data-center storage systems
2014	Orgerie *et al.* [41]	Energy efficiency of computing and network resources
2014	Mittal [42]	Energy efficiency in embedded computing systems
2014	Mittal *et al.* [43]	GPU Energy Efficiency
2014	Hammadi *et al.* [44], Bilal *et al.* [45][46]	Data center networks and their energy efficiency
2014	Ebrahimi *et al.* [47]	Data center cooling technology
2014	Rahman *et al.* [48], Mittal [49]	Data center power management
2014	Gu *et al.* [50]	VM power metering
2014	Kong *et al.* [51]	Renewable energy usage and/or carbon emission in data centers
2014	Shuja *et al.* [52]	Data center energy-efficiency

software whereas we focus on energy use in the entire data center.

Surveys conducted on network power consumption issues include the work done by Hammadi *et al.*, which focused on architectural evolution of data center networks (DCNs) and their energy efficiency [44]. Bilal *et al.* conducted a survey on data center networks research and described energy efficiency characteristics of DCNs [45], [46]. Shuja *et al.* surveyed energy efficiency of data centers focusing on the balance between energy consumption and quality of service (QoS) requirements [52]. Rahman *et al.* surveyed about power management methodologies based on geographic load balancing (GLB) [48]. Unlike our work, none of these surveys delve into the details on the construction of power models. Furthermore, they mostly only consider a single aspect of a data center. Another similar survey on power management techniques for data centers was presented by Mittal [49]. But again, their focus was not on modeling.

Recently several data center infrastructure level surveys have been conducted. For example, Ebrahimi *et al.* conducted a survey on the data center cooling technology, and discussed the power related metrics for different components in a data center in detail [47].

The remaining related surveys are interdisciplinary, and cover multiple aspects of data center power consumption. The survey conducted by Ge *et al.* focused on describing power-saving techniques for data centers and content delivery networks [39]. While achieving power savings is one application of the models we survey, our goals are broader, and we seek to survey general power modeling and prediction techniques. Orgerie *et al.* [41] surveyed techniques to improve the energy efficiency of computing and network resources, but did not focus on modeling and prediction. Gu *et al.* conducted a survey on power metering for virtual machines (VMs) in clouds [50]. But their work only focused on VM power models, where our work is more comprehensive and structured. Kong *et al.* [51] conducted a survey on renewable energy and/or carbon emission in data centers and their aim is different from the aim of this survey paper.

A chronologically ordered listing of the aforementioned surveys is shown in Table II. In this survey paper we study the existing literature from bottom up, from energy consumption at the digital circuit level on through to the data center systems of systems level. With this approach we can compare the energy consumption aspects of the data centers across multiple component layers. We believe that the bottom-up decompositional approach we follow as well as the comprehensive coverage of the literature on all components makes our work a unique contribution to the data center and cloud computing research communities.

III. DATA CENTER ENERGY CONSUMPTION: A SYSTEM PERSPECTIVE

In this section we describe how a data center is organized and the flow of electrical power within a typical data center. Later, we present an organizational framework to help readers design effective power models.

A. Power Consumption Optimization Cycle

Power flow and chilled water flow of an example data center is shown in Fig. 3 [54]. Data centers are typically energized through the electrical grid. However, there are also data centers which use diesel, solar, wind power, hydrogen (fuel cells), etc. among other power sources. The electric power from external sources (i.e., the total facility power) is divided between the IT equipment, the infrastructure facilities, and support systems by the switch gear. *Computer room air conditioning (CRAC)* units, a part of the cooling infrastructure, receive power through *uninterrupted power supplies (UPSs)* to maintain consistent cooling even during possible power failure. Note that certain power components such as flywheels or battery backup may not be available in many data centers. Fig. 3 acts as a model data center for most of the remaining parts of this paper.

An overall view of the framework used in this survey is shown in Fig. 4. In general we can categorize the constituents of a data center as belonging to one of two layers, software and

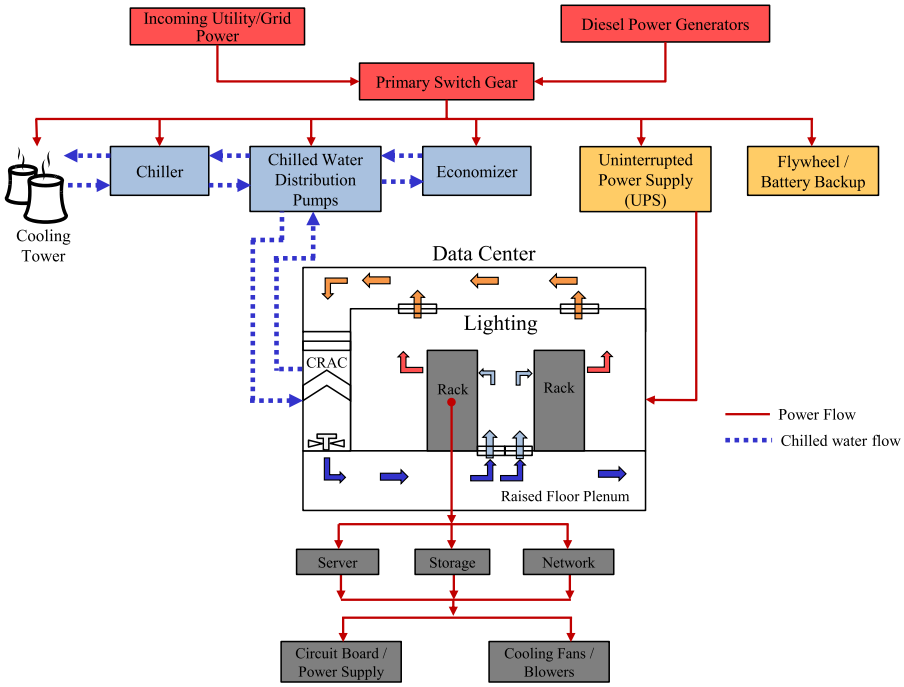


Fig. 3. Power flow in a typical data center [54]. Data centers are specifically designed to operate as server spaces and they have more control over their internal energy flow.

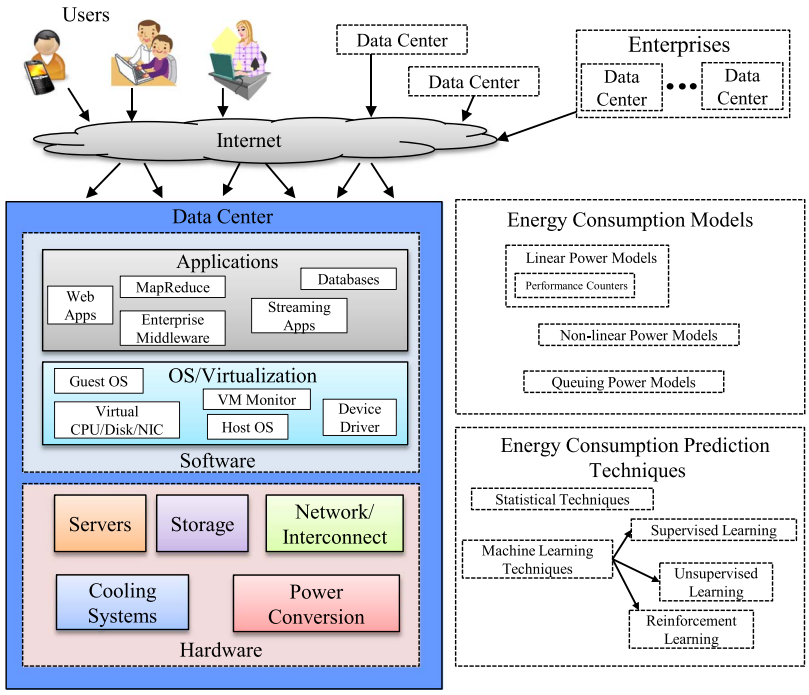


Fig. 4. A holistic view of the context for energy consumption modeling and prediction in data centers. The constituents of a data center can be categorized into two main layers: software layer and hardware layer.

hardware. The software layer can be further divided into two subcategories, the OS/virtualization layer, and the application layer. In the first half of this paper we describe the power consumption modeling work in the hardware layer. Later, we study power consumption modeling for software. Throughout this process, we highlight various energy consumption modeling and prediction techniques during this process which are applied at various different levels of the data center systems of systems.

Energy consumption optimization for such a complex systems takes the form of a *system optimization cycle*, as shown in Fig. 2. Modeling and prediction are two parts of this process.

Feature extraction constructs a model of the real world system to simulate its energy consumption. Feature extraction is often performed on miniature prototypes of the real world system (e.g., [55]), though it is also possible to conduct such extractions at the whole data center scale.

Raw power consumption measurements are one of the key inputs to this system optimization cycle. Many studies on the energy consumption of computer systems have been conducted using external power meters providing accurate measurements [56], [57] or using hardware or software instrumentation [58]. However, techniques that require the use of external meters or instrumentation are less portable because those require physical system access or invasive probing [59]. In the latest data center power metering techniques, power usage data is collected by polling power distribution units (PDUs) connected to IT equipment [60]. As mentioned earlier, PDUs are power strips that are created specifically to be used in data center environments. High-end intelligent PDUs offer per-socket measurements, rich network connectivity, and optional temperature sensors [61]–[63]. While dedicated PDU hardware provides accurate data on power consumption, present use of PDUs is costly and introduces system scalability issues. Hardware manufacturers are starting to deploy various sensors on HPC systems to collect power-related data as well as provide easier access to the gathered data. Modern internal on-board power sensors (e.g., the on-board power sensors on Tesla K20 GPUs [64]), power/thermal information reporting software such as AMESTER (IBM Automated Measurement of Systems for Temperature and Energy Reporting software) [65], HP Integrated Lights Out (iLO) [66] are several such examples. However, such facilities might not be available in many hardware platforms. Moreover, direct power measurement based energy consumption optimization techniques are rarely deployed in current data centers due to their usability issues. A more viable approach that has been widely used is to use hardware performance counters for energy consumption prediction.

Performance counters are the second type of input that can be used with the system optimization cycle. Performance counters are a special type of register exposed by different systems with the purpose of indicating their state of execution [67]. Performance counters can be used to monitor hundreds of different performance metrics such as cycle count, instruction counts for fetch/decode/retire, cache misses, etc. [68]. Performance counter information is used in many different tools and frameworks, alongside predefined power consumption models, for predicting the energy usage of systems. In certain situations performance counter based energy consumption modeling techniques can be augmented with physical power measurement. For example, Fig. 5 shows an approach for system power measurement that uses a combination of sampled multimeter data for overall total power measurements, and use estimates based on performance counter readings to produce per-unit power breakdowns [69].

The model construction process can be done by people as well as by computers using various intelligent techniques [70]. The model then needs to be validated to determine whether or not it is useful. In most of the power models presented in this paper, this step has been performed manually. However, there

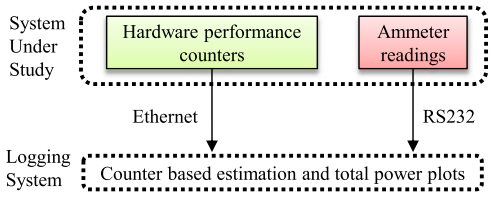


Fig. 5. A hybrid approach for system component level power consumption estimation [69]. This approach integrates measurements obtained from a multimeter with the performance counter readings to produce per-unit power breakdowns.

are situations where automatic model validation is done with the help of computers. Once validated the model can be used for different tasks such as prediction of the energy consumption of the data center. The experience gained by predicting the energy consumption of a real system can be utilized for improving the energy consumption model itself.

B. An Organizational Framework for Power Models

In this paper we map the energy consumption of a data center and its components to an *organizational framework*. We denote the instantaneous power dissipated at time t by,

$$P_t = f(\vec{S_t}, \vec{A_t}, \vec{E_t}). \quad (1)$$

The parameters in this equation are as follows,

- $\vec{S_t}$—represents the internal system state at time t. This can be further divided into three subcategories: physical, OS, and application software. Hardware configurations such as the type of processor, amount of memory, disk, and NIC structure are examples of the system state. Raw power measurements and performance counter values indicate the system status at a particular time.
- $\vec{A_t}$—represents input to the application at time t, including for example application parameters and input request arrival rates.
- $\vec{E_t}$—represents the execution and scheduling strategy [71] across the data center system at time t. Examples for scheduling include control of the CPU frequency, powering on or off the server, assignment of workload to different nodes or cores, etc. Which software we use at a particular time, how we configure the software stack, load balancing, and scheduling algorithms also determines the execution strategy.

The power model we use can either be additive in the power consumption of individual components, regression based, or use machine learning. The t value associated with each parameter denotes the time aspect of these parameters. However, in certain power models, e.g., the model in Equation (7), we do not observe the time of the measurements. In such cases the power is calculated using an average value over a time window. For simplicity, in the rest of the paper we simply use the parameter name, e.g., $\vec{A}$, instead of the time parameterized name, e.g., $\vec{A_t}$. This organizational model can be used for a number of

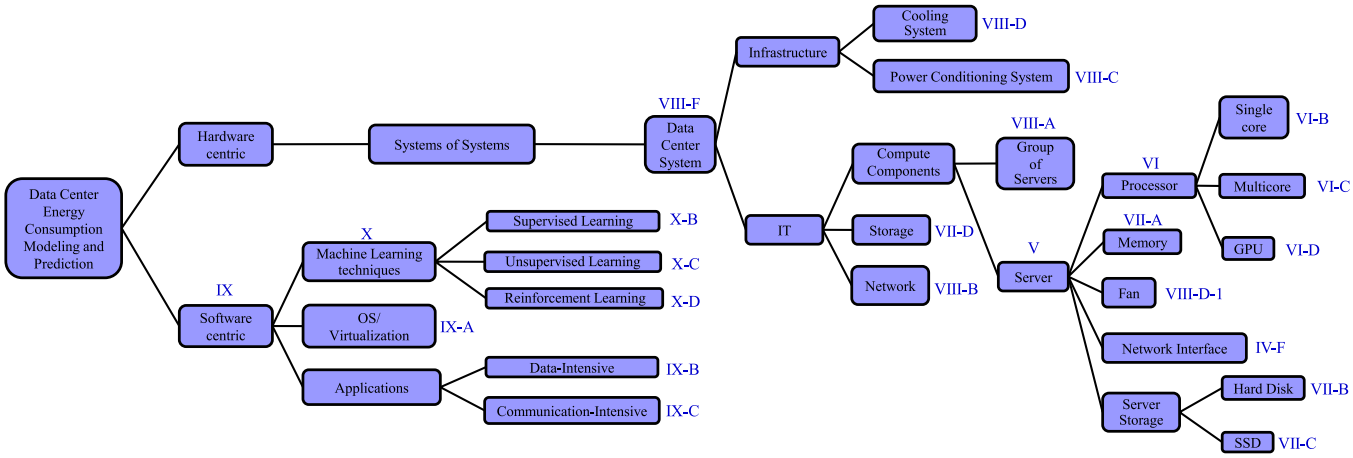


Fig. 6. A taxonomy based overview of the data center energy consumption modeling literature surveyed in this paper. Roman numbers and capital letters near the boxes indicate the section/subsection title in this paper.

purposes, including data center energy consumption prediction, data center system planning, system/subsystem comparisons, and energy consumption optimization. We refer to [72] for some example uses.

When the model is used for power consumption prediction at time t, it can be represented as follows,

$$P_{t+1} = g(\vec{S_t}, \vec{A_t}, \vec{E_t}), \tag{2}$$

where the function parameters are subscripted with $t = 0, 1, \ldots$, and the function g predicts the power usage of the next time step. If one knows the details of a system's physical state and the input for the system, she can schedule (i.e., adjust $\vec{E}$) the applications to operate most efficiently for a given power budget. In this case the applications' deployment schedule is determined by the other three parameters (P, $\vec{S}$, and $\vec{A}$). This general model of data center power consumption is reflected in many of the models described in this paper. However, the interpretation of the function f is different across different power models, as the models are based on different techniques and focus on different components. For example, power models such as the one given in Equation (10) may use a componentwise decomposition of power while models such as the one shown in Equation (4) use a static versus dynamic power decomposition.

A more detailed view of Fig. 4 is presented in Fig. 6. The latter figure provides an overview of the areas surveyed in this paper. We study models from two viewpoints, their level of abstraction and the techniques employed. The bounds of the abstractions follow the system boundaries as well as the application components of the data center system being modeled. Techniques described in this survey are of two types, either hardware centric or software centric. Software centric techniques can be further divided as performance counter based and machine learning based. In the subsequent sections, we first describe hardware centric techniques, starting from energy consumption modeling at the digital circuit level.

IV. Digital Circuit Level Energy Consumption Modeling

Power models play a fundamental role in energy-efficiency research of which the goal is to improve the components' and systems' design or to efficiently use the existing hardware [7]. Desirable properties of a full system energy consumption model include Accuracy (accurate enough to allow the desired energy saving), Speed (generate predictions quickly enough), Generality and portability (should be suitable for as many systems as possible), Inexpensiveness (should not requires expensive or intrusive infrastructure), and Simplicity [73]. This section provides an introduction to the energy consumption fundamentals in the context of electronic components.

A. Energy vs Power

Energy (E) is the total amount of work performed by a system over a time period (T) while power (P) is the rate at which the work is performed by the system. The relationship between these three quantities can be expressed as,

$$E = PT, \tag{3}$$

where E is the system's energy consumption measured in Joules, P is measured in *Watts* and T is a period of time measured in seconds. If T is measured in unit times then the values of energy and power become equal.

The above expression can be slightly enhanced by considering energy as the integration of power values in a time period starting from t_1 and ends at t_2. Note that we use the terms energy and power interchangeably in this paper.

B. Dynamic vs Static Power

Complementary metal-oxide semiconductor (CMOS) technology has been a driving force in recent development of computer systems. CMOS has been popular among the microprocessor designers due to its resilience for noise as well as low heat produced during its operation compared to other

semiconductor technologies. The digital CMOS circuit power consumption (P_{total}) can be divided into two main parts as,

$$P_{total} = P_{dynamic} + P_{static}, \tag{4}$$

where $P_{dynamic}$ is the dynamic power dissipation while P_{static} is the static power dissipation.

Dynamic power is traditionally thought of as the primary source of power dissipation in CMOS circuits [74]. The three main sources of dynamic power ($P_{dynamic}$) consumption in digital CMOS circuits are switched capacitance power (caused by the charging and discharging of the capacitive load on each gate's output), short-circuit power (caused by short-circuit current momentarily flowing within the cell), and leakage power (caused by leakage current irrespective of the gate's state) [75]. These power components can be represented as follows,

$$P_{dynamic} = P_{switching} + P_{short-circuit} + P_{leakage}, \tag{5}$$

where the first term $P_{switching}$ represents the switching component's (switching capacitance) power. The most significant component of power discharge in a well designed digital circuit is the switching component. The second term represents the power consumption that happens due to the direct-path short circuit current that occurs when both the N-type metal-oxide semiconductor (NMOS) and P-type metal-oxide semiconductor (PMOS) transistors are simultaneously active making the current directly flow to ground. The leakage current creates the third component which is primarily determined by the fabrication technology of the chip. In some types of logic styles (such as pseudo-NMOS) a fourth type of power called *static biasing power* is consumed [76]. The leakage power consists of both gate and sub-threshold leakages which can be expressed as [77],

$$\begin{cases} P_{leakage} = n_{gate} I_{leakage} V_{dd}, \\ I_{leakage} = AT^2 e^{-B/T} + Ce^{(r_1 V_{dd} + r_2)}, \end{cases} \tag{6}$$

where n_{gate} represents the transistor count in a circuit while $I_{leakage}$ represents the leakage current, and T corresponds to the temperature. The values A, B, C, r_1, and r_2 are constants. Circuit activities such as transistor switches, changes of values in registers, etc. contribute to the dynamic energy consumption [78].

The primary source of the dynamic power consumption is the switched capacitance (Capacitive power [79]). If we denote A as the switching activity (i.e., Number of switches per clock cycle), C as the physical capacitance, V as the supply voltage, and f as the clock frequency; the dynamic power consumption can be defined as in Equation (7) [80]–[82],

$$P_{capacitive} = ACV^2 f. \tag{7}$$

Multiple techniques are available for easy scaling of the supply voltage and frequency in large range. Therefore, the two parameters V and f attract a large attention by the power-conscious computing research.

Static power (P_{static}) is also becoming and important issue because the leakage current flows even when a transistor is switched off [78] and the number of transistors used in processors is increasing rapidly. Static power consumption of a transistor can be denoted as in Equation (8). Static power (P_{static}) is proportional to number of devices,

$$P_{static} \propto I_{static} V, \tag{8}$$

where I_{static} is the leakage current.

The above mentioned power models can be used to accurately model the energy consumption at the micro architecture level of the digital circuits. The validity of such models is a question at the higher levels of the system abstractions. However, as mentioned in Section I such basic power models have been proven to be useful in developing energy saving technologies. For example the power model described in Equation (7) makes the basis of dynamic voltage frequency scaling (DVFS) technique which is a state-of-the-art energy saving technique used in current computer systems.

V. Aggregate View of Server Energy Models

IT systems located in a data center are organized as components. Development of component level energy consumption models helps for multiple different activities such as new equipment procurement, system capacity planning, etc. While some of the discussed components may appear at different other levels of the data center hierarchy, all of the components described in this section are specifically attributed to servers. In this section we categorize the power models which provide aggregated view of the server power models as additive models, utilization based models, and queuing models.

A. Additive Server Power Models

Servers are the source of productive output of a data center system. Servers conduct most of the work in a data center and they correspond to considerable load demand irrespective of the amount of space they occupy [83]. Furthermore, they are the most power proportional components available in a data center which supports implementation of various power saving techniques on servers. In this sub section we investigate on the additive power models which represent the entire server's power consumption as a summation of its sub components. We follow an incremental approach in presenting these power models starting from the least descriptive models to most descriptive models. These models cloud be considered as an improvement over linear regression, where non-parametric functions are used to fit model locally and are combined together to create the intended power model [84].

One of the simplest power models was described by Roy *et al.* which represented the server power as a summation of CPU and memory power consumption [85]. We represent their power model as,

$$E(A) = E_{cpu}(A) + E_{memory}(A), \tag{9}$$

where $E_{cpu}(A)$ and $E_{memory}(A)$ are energy consumption of the CPU and the memory while running the algorithm A. More details of these two terms are available in Equations (54) and (87) respectively. Jain *et al.* have described a slightly different power model to this by dividing the energy consumption of

CPU and memory into two separate components as data and instructions [86].

More detailed power models have been created by considering other components of a server such as disks, network peripherals, etc. Server energy consumption model described by Tudor *et al.* [87] augments the above power model with I/O parameters. Their model can be shown as,

$$E_{total} = E_{cpu} + E_{memory} + E_{I/O}, \tag{10}$$

where energy used by the server is expressed as a function of energy used by CPU, memory, and I/O devices. However, most of the current platforms do not allow measuring the power consumed by the three main sub systems (CPU, Memory, and Disk) of servers separately. Only the full system power denoted by E_{total} can be measured [88]. Ge *et al.* have also described a similar power model by expressing the system power consumption as a summation of CPU, memory, and other system components [89]. The power model described in Equation (10) can be further expanded as [90],

$$E_{total} = E_{cpu} + E_{memory} + E_{disk} + E_{NIC}, \tag{11}$$

where E_{cpu}, E_{memory}, E_{disk}, and E_{NIC} correspond to energy consumed by CPU, memory, disk, and network interface card respectively. Furthermore, this model may incorporate an additional term for energy consumption of mother board as described in [91], [92] and in [93] or a baseline constant such as described in [94].

The above energy model can be further expanded considering the fact that energy can be calculated by multiplying average power with execution time as [90],

$$E_{total} = \overline{P}_{comp}T_{comp} + \overline{P}_{NIC}T_{comm} + \overline{P}_{net\_dev}T_{net\_dev}, \tag{12}$$

where $\overline{P}_{comp}$ denotes combined CPU and memory average power usage. T_{comp} is the average computation time. T_{comm} is the total network time and $\overline{P}_{NIC}$ is the average network interface card power. This energy model also takes into account the energy cost from network devices' power $\overline{P}_{net\_dev}$ and the running time T_{net_dev} when the devices are under load.

A slightly different version of this energy model can be constructed by considering the levels of resource utilization by the key components of a server [95] as,

$$P_t = C_{cpu,n}u_{cpu,t} + C_{memory}u_{memory,t} + C_{disk}u_{disk,t} + C_{nic}u_{nic,t}, \tag{13}$$

where u_{cpu} is the CPU utilization, u_{memory} is the memory access rate, u_{disk} is the hard disk I/O request rate, and u_{net} is the network I/O request rate. P_t refers to the predicted power consumption of server at time t while C_{cpu}, C_{memory}, C_{disk}, and C_{nic} are the coefficients of CPU, memory, disk and NIC respectively. This power model is more descriptive compared to the previously described server power models (in Equations (10) to (12)). System resource utilization values (u) can be regraded as a reflection of the job scheduling strategy of the modeled system. The more jobs get scheduled in the system, the CPU utilization increases accordingly).

In an almost similar power model, Lewis *et al.* described the entire system energy consumption using the following equation [96],

$$E_{system} = A_0(E_{proc} + E_{mem}) + A_1E_{em} + A_2E_{board} + A_3E_{hdd}, \tag{14}$$

where, A_0, A_1, A_2, and A_3 are unknown constants that are calculated via linear regression analysis and those remain constant for a specific server architecture. The terms E_{proc}, E_{mem}, E_{em}, E_{board}, and E_{hdd} represent total energy consumed by the processor, energy consumed by the DDR and SDRAM chips, energy consumed by the electromechanical components in the server blade, energy consumed by the peripherals that support the operation on board, and energy consumed by the hard disk drive (HDD). Use of single constant factor A_0 for both CPU and memory can be attributed to the close tie between CPU and memory power consumption.

CPU power consumption generally dominates the server power models [97]. This domination is revisited in multiple places of this survey. One example detailed system power model which possess this characteristic was described by Lent *et al.* where power consumption of a server is expressed as the sum of the power drawn by its sub components [98]. In this power model, the power (P) consumed by a network server hosting the desired services is given by,

$$P = I + \sum_{i=0}^{N-1}\alpha_N\rho_N(i) + \sum_{j=0}^{C-1}\alpha_C\rho_C(j) + \sum_{k=0}^{D-1}\alpha_D\rho_D(k) + \psi_m\left(\sum_{j=0}^{C-1}\rho_C(j)\right) + \psi_M\left(\sum_{j=0}^{C-1}\rho_C(j)\right), \tag{15}$$

where I denotes idle power consumption. Lent *et al.* assumed each of the subsystems will produce linear power consumption with respect to their individual utilization. Then the power consumption of a core, disk, or port subsystem can be estimated as the product of their utilization (core utilization ρ_C, disk utilization ρ_D, network utilization ρ_N) times constant factor (α_C, α_D, and α_N). These factors do not necessarily depend on the application workload. The model shown above does not have a separate subsystem for memory because the power consumed by memory access is included in the calculations of the power incurred by the other subsystems (especially by the core). CPU instruction execution tends to highly correlate to memory accesses in most applications [98]. The two components ψm and ψM are made to model behaviors that could be difficult to represent otherwise.

A different type of power models based on the type of operations conducted by a server can be developed as follows. In this approach which is similar to the power consumption of CMOS circuits described previously, computer systems' energy consumption (i.e., data center energy consumption) is divided into two components called static (i.e., baseline) power (P_{fix}) and dynamic (i.e., active) power (P_{var}) [75], [99], [100] which can be expressed as,

$$P_{total} = P_{fix} + P_{var}, \tag{16}$$

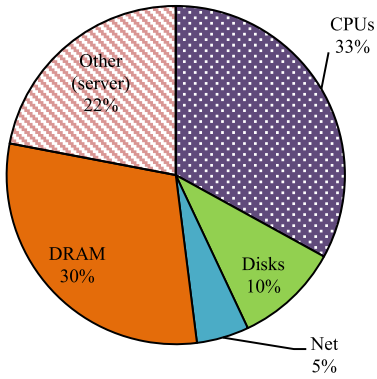


Fig. 7. An approximate distribution of peak power usage by components of a warehouse scale computer deployed at Google in 2007 [102].

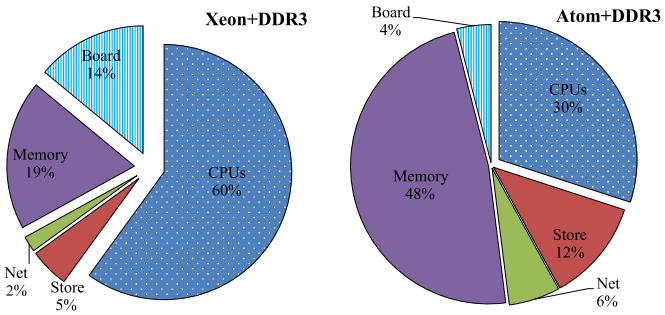


Fig. 8. Power breakdown across the components of two servers [103]. In the case of Atom processor based server, memory consumes largest amount of power while in Xeon based server, the CPUs are the main power consumers.

where the fraction between static and dynamic power depends on both the system under consideration and the workload itself.

Static power (P_{fix}) consumption in the context of a server is the power that is consumed by the system irrespective of its state of operation. This includes power wasted because of leaking currents in semiconductor components such as CPU, memory, I/O and other motherboard components, fans, etc. [101]. This category also includes power required to keep basic operating system processes and other idling tasks running (E.g., Power required to keep the hardware clocks, timer interrupts, network ports, and disk drives active [98]). The leaking currents need to be kept minimum to avoid such energy waste. However, this requires improvement of the lower level (semi-conductor chip level) energy consumption [75].

Dynamic power consumption in the context of a server is made by activities such as operation of circuits, access to disc drives (I/O), etc. It depends mainly on the type of workload which executes on the computer as well as how the workload utilizes the system's CPU, memory, I/O, etc. [101]. Furthermore, 30–40% of the power is spent on the disk, the network, the I/O and peripherals, the regulators, and the rest of the glue circuitry in the server.

Fig. 7 shows an example breakdown of power consumption of a server [102] deployed in a Google data center. It should be noted that the percentage power consumption among different components are not fixed entities. For example Fig. 8 shows power consumption comparison of two servers with one being a mobile processor (Atom processor) [103]. In the case of Atom processor based server, memory consumes the largest amount of power while in the Xeon based server, the CPUs are the main power consumers. The disk and the power supply unit are another two large contributors to this collection [104] which are specifically not shown in Fig. 8.

Note that most of the power models described in this subsection were based on component wise power consumption decomposition. However, there can be other different types of energy consumption models developed for a server based on its phases of execution. One such example is the energy model described by Orgerie *et al.* [105] (though its not specifically attributed to servers by them),

$$E = E_{boot} + E_{work} + E_{halt}, \tag{17}$$

where E_{boot} and E_{halt} corresponds to system booting and halting energy consumption which is zero if the equipment need not be booting or halting during its operation life cycle. However, use of this type of operation phase based energy models is quite rare in real world. On the contrary, system utilization based power models are heavily used in data center power modeling. We investigate this important are in the next subsection.

Another component wise power breakdown approach for modeling server power is use of the VM power as a parameter in the power model. This can be considered as an extension of the power model described in Equation (37). A power model which is based on this concept is described in [106] where the server power is expressed as,

$$P_{server} = P_{baseline} + \sum_{i=1}^{n} P_{vm}(i), \tag{18}$$

where P_{server} represents the total power of a physical server while $P_{baseline}$ is the baseline power that is empirically determined. P_{vm} is the power of an active VM, and n is the number of VMs held by the server. This power model can be further expanded by expressing the power usage of each and every VM. Each and every VM's power consumption can be expressed as in Equation (184). Then the complete server power can be expressed as,

$$P_{server} = \alpha \sum_{k=1}^{n} U_{cpu}(k) + \beta \sum_{k=1}^{n} U_{mem}(k) + \gamma \sum_{k=1}^{n} U_{io}(k) + ne + E_{baseline}, \tag{19}$$

where n is the number of VMs running in the physical node. A similar power model for server was created in [107] by considering only CPU, disk and idle power consumption. In that power model, CPUs and disks are considered as the major components that reflect the system activities.

B. System Utilization Based Server Power Models

The second main category of server power models has been created considering the amount of system resource utilization by its components. Traditionally the CPU has been the

largest power consumer in a server. Hence most of the system utilization based power models leverage CPU utilization as their metric of choice in modeling the entire system's power consumption. Different from previous subsection, we organized the content in this subsection in a chronological order because there is no clearly observable structural relationship between these power models.

One of the earliest server utilization based power models which appeared in year 2003 was an extension of the basic digital circuit level power model described in Equation (7). This was introduced by Elnozahy *et al.* where the fundamental dynamic power model described in Equation (7) was extended considering a simplification made on the system voltage [72], [108]. They expressed voltage as a linear function in the frequency where $V = \alpha f$ (α is a constant). This results in a power model for a server running at frequency f as,

$$P(f) = c_0 + c_1 f^3, \tag{20}$$

where c_0 is a constant that includes power consumption of all components except the CPU and the base power consumption of CPU. c_1 is a constant ($c_1 = AC\alpha^2$ where A and C are constants from Equation (7)).

In year 2006, Economou *et al.* described Mantis which is a non-intrusive method for modeling full-system power consumption and real-time power prediction. Mantis uses a one-time calibration phase to generate a model by correlating AC power measurements with user-level system utilization metrics [104]. Mantis uses the component utilization metrics collected through the operating system or standard hardware counters for construction of power models. Mantis has been implemented for two different server systems (highly integrated blade server and a Itanium server) for which the power models are depicted as,

$$\begin{aligned} P_{blade} &= 14.45 + 0.236u_{cpu} - (4.47E-8)u_{mem} \\ &\quad + 0.00281u_{disk} + (3.1E-8)u_{net}, \\ P_{itanium} &= 635.62 + 0.1108u_{cpu} + (4.05E-7)u_{mem} \\ &\quad + 0.00405u_{disk} + 0u_{net}, \end{aligned} \tag{21}$$

where the first term in both the above equations is a constant which represents the system's idle power consumption. Each u_{cpu}, u_{mem}, u_{disk}, and u_{net} correspond to CPU utilization, off-chip memory access count, hard disk I/O rate and network I/O rate respectively.

One of the notable processor utilization based power models is the work by Fan *et al.* (appeared in the year 2007) which has influenced recent data center power consumption modeling research significantly. Fan *et al.* have shown that the linear power model can track the dynamic power usage with a greater accuracy at the PDU level [6], [109]. If we assume the power consumed by a server is approximately zero when it is switched off, we can model the power P_u consumed by a server at any specific processor utilization u (u is a fraction [110]) as [6], [111], [112] in Equation (22),

$$P_u = (P_{max} - P_{idle})u + P_{idle}, \tag{22}$$

where P_{idle}, P_{max} are the average power values when the server is idle and the average power value when the server is fully utilized respectively. This model assumes server power consumption and CPU utilization has a linear relationship. Certain studies have used this empirical model as the representation of the system's total power consumption since Fan *et al.*'s study [109] have shown that the power consumption of servers can be accurately described by a liner relationship between the power consumption and CPU utilization [113].

The above processor utilization based power model has been highly influential in recent server power modeling research. For example, the works by Zhang *et al.* and Tang *et al.* used CPU utilization as the only parameter to estimate the system energy consumption [114], [115]. However, there are certain works which define slightly different utilization metric for the power model in Equation (22). In one such works [116], the power model appears in the context of modeling the energy consumption of a CDN server. Yet, in [116] the utilization metric has been changed as the percentage between the actual number of connections made to a server s against the maximum number of connections allowed on the server.

In the same work, Fan *et al.* also have proposed another empirical, non-linear power model as follows [73], [75], [109],

$$P_u = (P_{max} - P_{idle})(2u - u^r) + P_{idle}, \tag{23}$$

where r is a calibration parameter that minimizes the square error which needs to be obtained experimentally. In certain literature the value of r is calculated as 1.4 [109]. Fan *et al.* conducted an experiment which compared the accuracy of the power models in Equation (22) and Equation (23) using a few hundred servers in one of the Google's production facilities. Fan *et al.* mentioned that except for a fixed offset, the model tracks the dynamic power usage extremely well. The error was below 5% for the linear model and 1% for the empirical model. Although the empirical power model in Equation (23) had better error rate, one need to determine r calibration parameter which is a disadvantage associated with the model.

Several notable works of system utilization based server energy consumption modeling appeared in years 2010–2011. One such work was presented by Beloglazov *et al.* [111]. They considered the fact that CPU utilization may change over time due to the variation of the workload handled by the CPU [111]. Therefore, CPU utilization can be denoted as a function of time as [2], [117] in Equation (24),

$$E = \int_{t_0}^{t_1} P\left(u(t)\right) dt, \tag{24}$$

where E is the total energy consumption by a physical node during a time period from t_0 to t_1. $u(t)$ corresponds to the CPU utilization which is a function of time.

Multiple work have been done to model the aggregate power consumption of a server [112]. Wang *et al.* presented an energy model derived from experiments on a blade enclosure system [118]. They modeled server power as shown in the following equation,

$$P_{B_j} = g_B u_j + P_{B,idle}, \text{ for any blade j.} \tag{25}$$

They mentioned that CPU utilization (u) is a proxy for the effect of active workload management while the slope g_B and the intercept $P_{B,idle}$ captures the effect of power status tuning.

In their work on planning and deployment of enterprise applications, Li *et al.* conducted power consumption modeling of a server [119]. Unlike the previously described approaches, they used a normalized power unit P_{norm},

$$P_{norm} = \frac{P_{sys} - P_{idle}}{P_{busy} - P_{idle}}, \tag{26}$$

where P_{sys} is the system power consumption, P_{idle} is the idling power consumption (i.e., the utilization is zero, $(U = 0)$), P_{busy} is the power consumption when the system is completely utilized ($U = 1$). Furthermore, they described another model that relates normalized power (P_{norm}) and CPU utilization (U) as,

$$P_{norm}(U) = 1 - h(U)^{-1}, \tag{27}$$

where $h(U) = c_1 U^{c_2} + c_3 U^{c_4} + c_5$, while $(c_1, \ldots, c_5)$ are parameters to be fitted.

In an effort to build on the power model in Equation (23), Tang *et al.* created somewhat sophisticated power consumption model [120] as,

$$P_x(t) = P_{x\_idle} + (P_{x\_full} - P_{x\_idle})\alpha_x U_x(t)^{\beta_x}, \tag{28}$$

where P_{x_idle} and P_{x_full} are the power consumption of a server x at idle and fully loaded states respectively. α_x and β_x are server dependent parameters. U_x corresponds to the CPU utilization of server x at time t. The notable difference from the power model in Equation (23) is the addition of temporal parameter to the power model and the feature of accounting multiple different servers.

Yao *et al.* described the power consumption of a server as follows [121],

$$P = \frac{b_i(t)^{\alpha}}{A} + P_{idle}, \tag{29}$$

where A, P_{idle}, and α are constants determined by the data center. P_{idle} is the average idle power consumption of the server. $b_i(t)$ denotes the utilization/rate of operation if the server i at time t. Yao *et al.* selected the values $\alpha = 3$, $P_{idle} = 150$ Watts, and A such that the peak power consumption of a server is 250 Watts. Both the power models in Equations (28) and (29) were developed in the year 2011.

A similar power model for a server i was made by Tian *et al.* [122] in year 2014. However, they replaced the frequency parameter with service rate($\mu_i^{\alpha_i}$) and utilization of server u_i as,

$$P_i = u_i k_i \mu_i^{\alpha_i} + P_i^*, \tag{30}$$

where P_i^* represents the static power consumption of server i. This type of power models are also known as Power-Law models in certain literature [123]. It can be observed that when considering one decade period from the year 2003, many of the utilization based power models have appeared around 2010–2011 period. This indicates there are many recent work being carried out in this area.

Certain power models consider the System's CPU die temperature along with the CPU utilization to calculate the heat generated by the server. In a steady-state such heat dissipated by the server can be equated to the server power dissipation. In one such work the power dissipation by a server (P_{server}) is given by a curve-fitting model [124],

$$P_{server} = P_{IT} + P_{sfan}, \tag{31}$$

where P_{IT} represents the server heat generation excluding the heat generation by server cooling fan (P_{sfan}). The component of the above model can be expanded as,

$$P_{IT} = 1.566 \times 10^{-5} + 42.29u + 0.379T + 0.03002T^2, \tag{32}$$

where the R^2 value of the curve-fitting line was 0.9839. T is the CPU die temperature and u is the CPU utilization.

Furthermore, in certain works [125] the server's CPU usage and operation frequency are used for modeling a server's power consumption. The work by Horvath *et al.* is an example where they expressed the server power consumption as,

$$P_i = a_{i3} f_i u_i + a_{i2} f_i + a_{i0}, \tag{33}$$

where p_i, f_i, u_i represent the power consumption, processor's frequency, and utilization of node i respectively. $a_{ij} (j = 0, 1, 2, 3)$ are system parameters which can be determined by using the system identification of the physical server. They used the stead-state result of the M/M/n queuing model. The node utilization u is described as $u = \frac{x}{sn}$ where x is the number of concurrent tasks in current sampling cycle. Arrival rate s is the number of served tasks and n is the server's core count. When constructing the server power model they assumed that all servers are homogeneous with parameters $a_{i3} = 68.4$, $a_{i2} = 14.6$, $a_{i1} = -14.2$, $a_{i0} = 15.0$.

C. Other Server Power Models

Additive and utilization based power models represent majority of the server power models. However, there are multiple other power models which cannot be specifically attributed to these two categories. This sub section investigates on such power models. We follow a chronological ordering of power models as done in the previous section.

In a work on operational state based power modeling, Lefurgy *et al.* have observed that server power consumption changes immediately (within a millisecond) as the system's performance state changes from irrespective of the previous performance state [126]. Therefore, they concluded that power consumption of a server for a given workload is determined solely by the performance settings and is independent of the power consumption in previous control periods. Furthermore, from the performance experiments conducted by Lefurgy *et al.*, it was observed that a linear model fits well with an $R^2 > 99\%$ for all workloads. Therefore, they proposed a server power model as,

$$p(k) = At(k) + B, \tag{34}$$

where A and B are two system dependent parameters. $p(k)$ is the power consumption of the server in the k^{th} control period

while $t(k)$ is the performance state of the processors in the kth control period. Furthermore, they created a dynamic model for power consumption as,

$$p(k+1) = p(k) + Ad(k). \tag{35}$$

In a detailed work on server power modeling with use of regression techniques, Costa *et al.* [127] introduced a model for describing the power consumption of a computer through use of a combination of system wide variables (i.e., system wide metrics such as *host_disk.sda_disk_time_write* (average time a write operation took to complete), *host_cpu.X_cpu.system_value* (processes executing in kernel mode), etc.) Y_i, $i = 1, \ldots, I$; variables X_{jl}, $j = 1, \ldots, J$ describing individual process P_l, $l = 1, \ldots, L$. They took the power consumption of a computer with no load be denoted by P_0, and the respective coefficients of the regression model be called α_i for system wide variables, and β_j for per process variables. Based on the above definitions, the power consumption P of a computer can be denoted as,

$$P = P_0 + \sum_{i=1}^{I} \alpha_i Y_i + \sum_{j=1}^{J} \beta_j \sum_{l=1}^{L} X_{jl}. \tag{36}$$

Regression based power modeling has been shown to perform poorly on non-trivial workloads due to multiple reasons such as, level of cross dependency present in the features fed to the model, features used by previous approaches are outdated for contemporary platforms, and modern hardware components abstract away hardware complexity and do not necessarily expose all the power states to the OS. The changes in the power consumption are not necessarily associated with changes in their corresponding states [128].

Queuing theory has been used to construct server power models. In one such work, Gupta *et al.* created a model for power consumption of a server [113]. They assumed that servers are power proportional systems (i.e, assuming server power consumption and CPU utilization has a linear relationship) [113]. They described the power consumption of a server as,

$$P(\lambda) = \frac{\lambda}{\mu}(P_{cpu} + P_{other}) + \left(1 - \frac{\lambda}{\mu}\right) P_{idle}, \tag{37}$$

where P_{cpu} and P_{other} represent the power consumption of the processor and other system components while P_{idle} is the idle power consumption of the server. They assumed that the processor accounts for half of the system power during active periods and the system consumes 10% of its peak power during idle periods. They used queuing theoretic models for capturing the request processing behavior in data center servers. They used the standard M/M/1 queuing model which assumes exponentially distributed request inter-arrival time with mean $\frac{1}{\lambda}$ and an exponentially distributed service time with mean $\frac{1}{\mu}$.

In a work conducted in year 2012, Enokido *et al.* created Simple Power Consumption (SPC) model for a server s_t where the power consumption rate $E_t(\tau)$ at time τ is given by [129],

$$E_t(\tau) = \begin{cases} R_t, & \text{if} \quad Q_t(\tau) \geq 1, \\ \min E_t, & \text{otherwise}, \end{cases} \tag{38}$$

where R_t shows the maximum power consumption rate where a rotation speed of each sever fan is fixed to be minimum. In the SPC model if at least one process p_i is performed, the electric power is consumed at fixed rate R_t on a server s_t at time τ ($E_t(\tau) = R_t$). If not the electric power consumption rate of the server s_t is minimum.

Furthermore, they created an extended power model for a server considering the power consumption of cooling devices (i.e., fans) [129]. They did not consider how much electronic power each hardware component of a server like CPU, memory, and fans consume. They rather considered aggregated power usage at macro level. In their *Extended Simple Power Consumption (ESPC)* model, $E_t(\tau)$ shows the electric power consumption rate [W] of a server s_t at time $\tau (t = 1, \ldots, n)$, $\min E_t \leq E_t(\tau) \leq \max E_t$ (See Equation (39)). Different from the model described in Equation (112) they used an additional parameter R_t in this model. Then the ESPC is stated as,

$$E_t(\tau) = \begin{cases} \max E_t, & \text{if} \quad Q_t(\tau) \geq M_t, \\ \rho_t . Q_t(\tau) + R_t, & \text{if} \quad 1 \leq Q_t(\tau) \leq M_t, \\ \min E_t, & \text{otherwise}, \end{cases} \tag{39}$$

where ρ_t is the increasing ratio of the power consumption rate on a server s_t. $\rho_t \geq 0$ if $Q_t(\tau) > 1$ and $\rho_t = 0$ if $Q_t(\tau) = 1$.

In another system utilization based energy consumption model by Mills *et al.* the energy consumed by a compute node with CPU (single) executing at speed σ is modeled as [130],

$$E(\sigma, [t_1, t_2]) = \int_{t_1}^{t_2} \left(\sigma^3 + \rho \sigma_{max}^3\right) dt, \tag{40}$$

where ρ stands for overhead power which is consumed regardless the speed of the processor. The overhead includes the power consumption by all other system components such as memory, network, etc. Although the authors mentioned the energy consumption of a socket, their power model is generalized to the entire server due to this reason.

In certain works power consumption of a server is calculated by following a top down approach, by dividing the total power consumption of servers by the number of servers hosted in the data center [131]. However, such power models are based on a number of assumptions such as uniform server profiles, homogeneous execution of servers, etc.

Certain power consumption modeling techniques construct metrics to represent the energy consumption of their target systems. In one such work, Deng *et al.* defined a metric called *system energy ratio* to determine the best operating point of a full compute system [133]–[135]. For a memory frequency of f_{mem} they defined the system energy ratio (K) as,

$$K(f_{mem}) = \frac{T_{f_{mem}} P_{f_{mem}}}{T_{base} P_{base}}, \tag{41}$$

where $T_{f_{mem}}$ corresponds to the performance estimate for an epoch at frequency f_{mem}. On the otherhand $P_{f_{mem}} = P_{mem}(f_{mem}) + P_{nonmem}$, where $P_{mem}(f)$ is calculated according to the model for memory power for Micron DDR SDRAM [136]. P_{nonmem} accounts for all non-memory subsystem components. The corresponding values for the $T_{f_{mem}}$ and $P_{f_{mem}}$

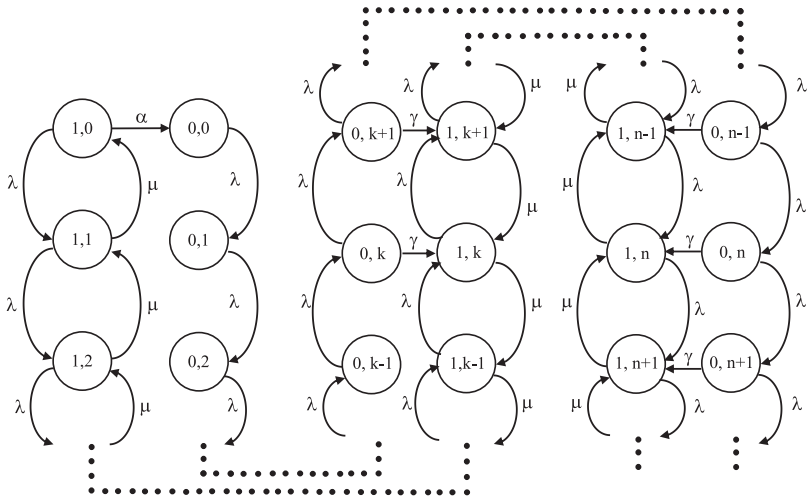


Fig. 9. $M/M/1 \circ M, M, k$ queue Markov chain [132] representation of a server. The model allows one to determine an optimal policy for a single server system under a broad range of metrics.

at a nominal frequency are denoted as T_{base} and P_{base}. Note that when considering the multiple memory frequencies $(f_{mc_1}, f_{mc_2}, \ldots, f_{mc_n})$ the term $P_{f_{mem}}$ can be expanded further as $P_{f_{mem}} = \sum_i P_{f_{mc_i}} + P_{nonmem}$ [134]. The definition of K was further expanded by Deng *et al.* as,

$$K\left(f^1_{core}, \ldots, f^n_{core}, f_{mem}\right) = \frac{T_{f^1_{core},\ldots f^n_{core} f_{mem}} P_{f^1_{core},\ldots f^n_{core} f_{mem}}}{T_{base} P_{base}}, \tag{42}$$

where T_{base} and P_{base} are time and average power at a nominal frequency (e.g., maximum frequencies). Furthermore, they expressed the full system power usage as,

$$P\left(f^1_{core}, \ldots, f^n_{core}, f_{mem}\right) = P_{non} + P_{cache} + P_{mem}(f_{mem}) + \sum_{i=1}^{n} P^i_{core}\left(f^i_{core}\right), \tag{43}$$

where P_{non} is the power consumed by all system components except cores, the shared L2 cache, and the memory subsystem. P_{non} is assumed to be fixed. The average power of the L2 cache is denoted by P_{cache} and is calculated by using the cache's leakage and the number of accesses during the epoch. $P_{mem}(f)$ is the average power of L2 misses (which makes the CPU to access memory) and is calculated based on the description given in [137]. The value of $P^i_{core}(f)$ is calculated based on the core's activity factor following the same techniques used by [18] and [69]. Deng *et al.* used several L1 and L2 performance counters (Total L1 Miss Stalls (TMS), Total L2 Accesses (TLA), Total L2 Misses (TLM), and Total L2 Miss Stalls (TLS)) and per-core sets of four Core Activity Counters (CAC) which track committed ALU instructions, FPU instructions, branch instructions, and load/store instructions to estimate core power consumption [135]. They found that power usage of cores is sensitive to the memory frequency.

Power consumption states can be used for construction of server power models. Maccio *et al.* described an energy consumption model for a server by mapping its operational state into one of the four system states: *LOW*, *SETUP*, *BUSY*, and *IDLE* where each of the power states are denoted by E_{LOW}, E_{SETUP}, E_{BUSY}, and E_{IDLE}. They modeled the server power consumption as a Markov chain [132] (shown in Fig. 9). The state (n_1, n_2) means that the server is off when $n_1 = 0$ and on when $n_1 = 1$. There are n_2 jobs in the system. The model allows one to determine an optimal policy for a single server system under a broad range of metrics which considers the expected response time of a job in the system, the expected energy consumed by the system, and the expected rate that the server switches between the two energy states (off/on).

We described the research conducted on modeling the energy consumption of a data center server (as a complete unit) upto this point. All of these models are linear models while some of them listed in equations (9) , . . . , (17) are componentwise breakdown of the power of processors. The server power models in equations (22)–(24), and (28) are examples for non-linear models and those are based on CPU utilization. The power model in Equation (37) is different from the rest of the power models since it utilized queuing theory. A summary of the server power consumption models is shown in Table III. Many of the aforementioned power models denote a server's energy consumption as a summation of the power drawn by its subcomponents. In the next two sections (Sections VI and VII) of this paper we conduct a detailed investigation of the attempts made to model the power consumption of these sub components.

VI. Processor Power Models

Today, CPU is one of the largest power consumers of a server [6]. Modern processors such as Xeon Phi [138] consists of multiple billions of transistors which makes them utilize huge amount of energy. It has been shown that the server power consumption can be described by a linear relationship between the power consumption and CPU utilization [139]. CPU frequency to a large extent decides the current power utilization of a processor [140]. Comprehensive CPU power consumption models rely on specific details of the CPU microarchitecture and achieve high accuracy in terms of CPU power

TABLE III
SUMMARY OF SERVER POWER CONSUMPTION MODELING APPROACHES

Work(s)	Characteristics	Limitations
[85][86][87][92] [95][96][98]	Component wise breakdown of server power.	Depends on multiple assumptions.
[90]	Component wise breakdown of server power considering temporal features.	Needs accurate techniques to measure the time averages.
[106]	Component wise breakdown of server power considering power consumption of each VM run by the server.	Needs to measure the energy usage of each subcomponents. Also have to calculate the values of α, β, γ, and e.
[75][99][100][105]	Breakdown of server power considering state of operation.	Needs accurate techniques to measure the time averages.
[6][111][112]	Widely used linear power model.	Depends on multiple assumptions.
[73][75][109]	Non-linear. The value of r need to be known in advance.	Depends on multiple assumptions.
[2][117]	Non-linear power model based on mathematical integration. Addresses temporal aspects of the processor power.	Depends on multiple assumptions.
[120]	Non-linear power model. Addresses temporal aspects of the processor power.	α_x and β_x parameters need to be known beforehand.
[113][132]	Non-linear. Based on queuing theory. Request arrival rate and service time are considered.	Request arrival rates and service times need to be known beforehand.
[118]	Non-linear. Based on the level of system CPU utilization.	Depends on multiple assumptions.
[119]	Relates CPU utilization with system power consumption.	Based on multiple assumptions.
[127]	Regression based model. Considers both system wide variables as well as per process variables	Need to know α and β variables beforehand.
[129]	Non-linear. Considers the cooling power consumption of a server.	Depends on multiple assumptions.
[133][134][135]	Non-linear. Defines a metric called *system energy ratio*.	Depends on multiple assumptions.

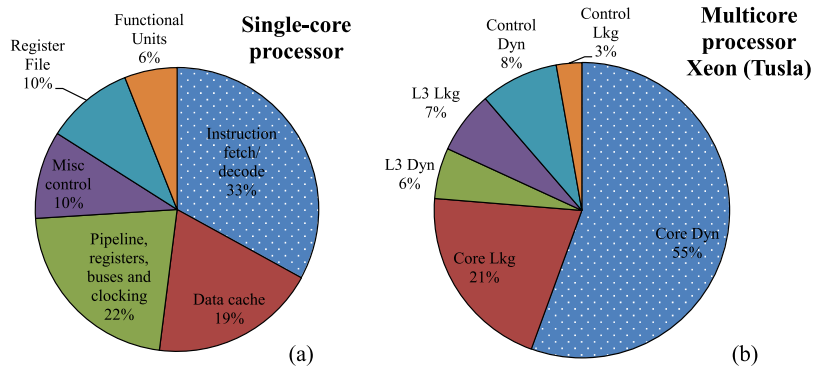


Fig. 10. Power breakdown of (a) single-core [142] and (b) multicore processors [143]. Note that leakage power is significant portion in the multicore processor.

consumption modeling [69], [141]. This section first provide a general overview for processor power modeling. Next, it delves into the specific details of modeling power consumption of three major categories of processors used in current data center systems: Single-core CPUs, Multicore CPUs, and GPUs.

A. Processor Power Modeling Approaches

Similar to system power consumption, processor power consumption can be modeled at very high level as *static* and *dynamic* power consumption. When analyzing the ratio between the static power and dynamic power it has been observed that processors presenting a low or very low activity will present a too large static power compared to dynamic power. Importance of the leakage and static power increases in very deep submicron technologies. Hence static power has become an important concern in recent times. A real world example of this phenomenon can be observed in the power breakdown of two single-core and multicore processors shown in Fig. 10 [142], [143]. While the two charts are based on two types of categorizations it can be clearly observed that multicore processor has significant amount of power leakage. Another observation to note is that processor caches contribute to a significant percentage of processor power consumption. While not shown in Fig. 10, caches in IBM POWER7 processor consumes around 40% of the processor's power [144].

Two high level approaches for generating power models for processors are circuit-level modeling (described in Section IV) and statistical modeling (related techniques are described in Sections IX and X). Circuit-level modeling is a more accurate but computationally expensive approach (for example it is used in Wattch CPU power modeling framework [81]). Statical modeling on the other hand has a high up-front cost while the model is being trained, but this technique is much faster to use [145].

Statistical approaches for processor power modeling [18], [141], [146] are based on the data analysis conducted on the processor performance counters. Micro architectural events for the performance measurement purpose can be obtained from most modern microprocessors. Heuristics can be selected from the available performance counters to infer power relevant events and can be fed to an analytical processor to calculate

the power. Multiple different power modeling tools being developed based on CPU performance counters. *Virtual Energy Counters (vEC)* is an example tool that provides fast estimation of the energy consumption of the main components of modern processors. The power analysis is mainly based on the number of cache references, hits, misses, and capacitance values. vEC can address this problem but results in loss of coverage.

However, certain works have expressed the negative aspects of use of performance counters for power modeling. Economou *et al.* have pointed out that performance monitoring using only counters can be quite inaccurate since most processors allow for the measurement of only a limited number of concurrent counter readings [104]. Processor counters provide no insight into the I/O system such as disk and networking which makes it difficult to create accurate processor power models. According to [141] performance counters can provide good power estimation results and they have estimated power within 2% of the actual power. According to Jarus *et al.* and Costa *et al.* the average error lies less than 10% range [127], [147]. However, in general the availability of heuristics is limited by the types of the performance counters and the number of events that can be measured simultaneously [146].

B. Power Consumption of Single-Core CPUs

While we are living in a multicore era, it is better to investigate on the single-core power models first, because many of the current multicore processor power models are reincarnations of single-core power models. This subsection is organized as two main parts. In the first half we describe the additive power modeling techniques of single-core processors. In the second half we investigate of the use of performance counters for power modeling of single-core processors.

Additive power modeling efforts are present in the context of single-core CPUs. One such power model was described by Shin *et al.* [148] as,

$$P_{cpu} = P_d + P_s + P_0, \tag{44}$$

where P_d, P_s and P_0 correspond to dynamic, static, and always-on power consumption. The dynamic power is expressed using the Equation (7). They mentioned that it is sufficient to include the two major consumers of leakage power in the static power model, which are *subthreashold leakage* and *gate leakage* power. Since the static power consumption is also dependent on the die temperature they incorporated T_d as a term in their power model which is expressed as,

$$P_s(T_d) = V_{dd}\left(K_1 T_d^2 e^{\frac{K_2 V_{dd}+K_3}{T_d}} + K_4 e^{(K_5 V_{dd}+K_6)}\right), \tag{45}$$

where K_n is a technology constant. They then expanded the right-hand side of the equation as a Taylor series and retained its linear terms as,

$$\begin{aligned} P_s(T_d) &= \sum_{n=0}^{\infty}\left(\frac{1}{n!}\right)\frac{d^n P_s(T_r)}{dT_d^n}(T_d - T_r)^n, \\ &\approx P_s(T_r) + \frac{dP_s(T_r)}{dT_d}(T_d - T_r), \end{aligned} \tag{46}$$

where T_r is a reference temperature, which is generally some average value within the operational temperature range.

Additive power models for single core CPUs can be created by aggregating the power consumption of each architectural power components which comprises of the CPU. Following this approach, Bertran *et al.* expressed the total power consumption of a single core CPU [149] as,

$$P_{total} = \left(\sum_{i=1}^{i=n} A_i \times P_i\right) + P_{static}, \tag{47}$$

where the weight of component i is represented as P_i and the activity ratio of the component i is represented as A_i while there are total n components n the CPU. The dynamic power consumption of component i is represented by $A_i \times P_i$, while P_{static} represents the overall static power consumption of all components. In their case study they used an Intel Core 2 Duo processor and they identified more than 25 microarchitectural components. For example, in their modeling approach they divided the whole memory subsystem into three power components: L1 and L2 caches and the main memory (MEM) (which includes the front side bus (FSB)). They also defined INT, FP, and SIMD power components which are related to the out-of-order engine of Core 2 Duo processor. Note that definition of a power component has been a challenge faced by them since certain microarchitectural components are tightly related to each other. Furthermore, there are certain microarchitectural components that do not expose any means of tracing their activities level.

As mentioned in the previous subsection, complete system power consumption can be measured online through microprocessor performance counters [18], [150]. Counter-based power models have attracted a lot of attention because they have become a quick approach to know the details of power consumption [151]. Bircher *et al.* showed that well known performance related events within a microprocessor (e.g., cache misses, DMA transactions, etc.) are highly correlated to power consumption happening outside the microprocessor [152]. Certain studies have used the local events generated within each subsystem to represent power consumption. Through such performance counter based energy consumption prediction, the software developers are able to optimize the power behavior of an application [153].

One of the earliest notable efforts in this area, Isci *et al.* described a technique for a coordinated measurement approach that combines real total power measurement with performance-counter-based, perunit power estimation [69]. They provided details on gathering live, per-unit power estimates based on hardware performance counters. Their study was developed around strictly co-located physical components identifiable in a Die photo. They selected 22 physical components and used the component access rates to weight the component power numbers. If each hardware component is represented as C_i the power consumption of a component $P(C_i)$ can be represented as,

$$P(C_i) = A(C_i)S(C_i)M(C_i) + N(C_i), \tag{48}$$

where $A(C_i)$ corresponds to the access counts for component C_i. $M(C_i)$ is the maximum amount of power dissipated by

each component. The $M(C_i)$ value is estimated by multiplying the maximum power dissipation of die by the fraction of area occupied by the component. $S(C_i)$ corresponds to a scaling factor introduced to scale the documented maximum processor power by the component area ratios. $N(C_i)$ corresponds to fixed power dissipation made by each component. The total power of the processor (P_{total}) is calculated as,

$$P_{total} = \sum_{i=1}^{22} P(C_i) + P_{idle}. \quad (49)$$

where the total power P_{total} is expressed as the sum of the power consumption of the 22 components and the idle power (P_{idle}). Note that in Equation (48) $M(C_i)$ and $S(C_i)$ terms are heuristically determined, $M(C_i)$ is empirically determined by running several training benchmarks that stress fewer architectural components and access rates are extracted from performance counters [154].

Processor cache hit/miss rates can be used to construct simple power model. In such power model described by Shiue *et al.*, the processor energy consumption is denoted as [155],

$$E = R_{hit}E_{hit} + R_{miss}E_{miss}, \quad (50)$$

where E_{hit} is the sum of energy in the decoder and the energy in the cell arrays, while E_{miss} is the sum of the E_{hit} and the energy required to access data in main memory.

In another similar work Contreras *et al.* used instruction cache misses and data dependency delay cycles in the Intel XScale processor for power consumption estimation [156]. Assuming a linear correlation between performance counter values and power consumption they use the following model to predict the CPU power consumption (P_{cpu}),

$$P_{cpu} = A_1(B_{fm}) + A_2(B_{dd}) + A_3(B_{dtlb}) + A_4(B_{itlb}) + A_5(B_{ie}) + K_{cpu}, \quad (51)$$

where $A_1, \ldots, A_5$ are linear parameters (i.e., power weights) and K_{cpu} is constant representing idle processor power consumption. The performance counter values of instruction fetch miss, number of data dependencies, data TLB misses, instructions TLB misses, number of instructions executed (i.e., *InstExec*) are denoted by $B_{fm}, B_{dd}, B_{dtlb}, B_{itlb}$, and B_{ie} respectively. However, they mentioned that in reality non-linear relationships exist.

While modern CPUs offer number of different performance counters which could be used for power modeling purposes its better to identify some key subset of performance counters that can better represent the power consumption of a CPU. One work in this line is done by Chen *et al.* where they found that five performance counters are sufficient to permit accurate estimation of CPU power consumption after conducting experiments with different combinations of hardware performance counters [157]. The five performance counters are,

1) number of L1 data cache references per second (α),
2) number of L2 data cache references per second (β),
3) number of L2 data cache references per second (γ),
4) number of floating point instructions executed per second (η), and
5) number of branch instructions retired per second (θ).

After assuming each access to system components such as L1, L2 caches consumes a fixed amount of energy, a power model for CPU can be created as follows,

$$P = b_0 + b_1\alpha + b_2\beta + b_3\gamma + b_4\eta + b_5\theta + b_6 f^{1.5}, \quad (52)$$

where f is the CPU frequency of which the exponent 1.5 was determined empirically. b_i, $i = 0, \ldots, 6$ are task-specific constants that can be determined during pre-characterization. b_0 represents the system idle and leakage power.

Merkel *et al.* [158] constructed a power model for processors based on events happening in the processor. They assumed that processor consumes a certain fixed amount of energy for each activity and assign a weight to each event counter that represents the amount of energy the processor consumes while performing the activities related to that specific activity. Next, they estimated the energy consumption of the processor as a whole by choosing a set of n events that can be counted at the same time, and by weighting each event with its corresponding amount of energy α_i. Therefore, they determine the amount of energy the processor consumes during a particular period of time by counting the number of events that occur during that period of time as follows,

$$E = \sum_{i-1}^{n} \alpha_i c_i. \quad (53)$$

Another performance counter based power model for CPUs was introduced by Roy *et al.* They described the computational energy ($E_{cpu}(A)$) consumed by a CPU for an algorithm A as follows [85],

$$E_{cpu}(A) = P_{clk}T(A) + P_w W(A), \quad (54)$$

where P_{clk} is the leakage power drawn by the processor clock, $W(A)$ is the total time taken by the non-I/O operations performed by the algorithm, $T(A)$ is the total time taken by the algorithm, and P_w is used to capture the power consumption per operation for the server. Note that the term "operation" in their model simply corresponds to an operation performed by the processor.

It can be observed that the complexity of the performance counter based power consumption models of single core processors have increased considerably over a decade's time (from year 2003 to 2013). Furthermore, it can be observed that a number of performance counter based power models have appeared in recent times.

C. Power Consumption of Multicore CPUs

Most of the current data center system servers are equipped with multicore CPUs. Since the use of different cores by different threads may create varied levels of CPU resource consumption, it is important to model the energy consumption of a multicore CPU at the CPU core level. Server's power consumption depends on the speed at which the core works. A high level

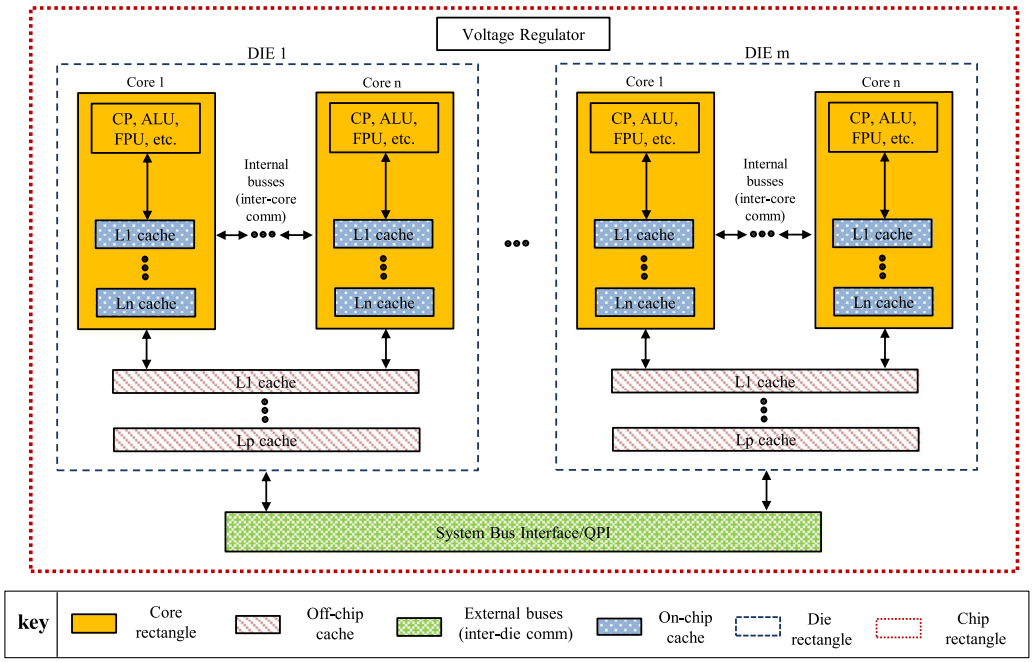


Fig. 11. An abstract architecture of a multicore processor [159]. All the components that lie within core-level rectangles are limited to specific core and cannot be shared with other cores.

architecture of a multicore processor is shown in Fig. 11 [159]. It consists of several dies with each one having several cores. Notable power consumers inside a processor include ALU, FPU, Control Unit, On/Off-chip Cache, Buses which are shown in Fig. 11. Cores are highlighted in yellow color while dies are denoted with dashed lines. Currently there are two main approaches for modeling the power consumption of a multicore processor: *queuing theory* and *component wise breakdown* of the power consumption. We will first describe the multicore CPU power models constructed using queuing theory. Next, we will move on to listing the power models constructed via component-wise breakdown/additive power modeling.

The work by Li *et al.* is an example of use of the first approach (queuing theory) for modeling the power consumption of multicore processors. They treated a multicore CPU as an M/M/m queuing system with multiple servers [160]. They considered two types of core speed models called idle-speed model (a core runs at zero speed when there is no task to perform) and constant-speed model (all cores run at the speed s even if there is no task to perform) [161]. Such constant speed model has been assumed in several other work such as [149] which helps to reduce the complexity of the power model. When constant speed model is employed for modeling the power consumption of a multicore processor, the processor's total energy consumption can be expressed as [159],

$$P_n = \sum_{j=1}^{n} P_c(j), \tag{55}$$

where P_n denotes the power consumption of n cores and $P_c(j)$ corresponds to the power dissipation of a core j. Power consumption of a single core $P_c(j)$ can be further described using a power model such as the ones described in the previous section (Section VI-B).

One of the simplest forms of denoting the maximum energy consumption of a core (E_{max}) is as follows,

$$E_{max} = D_{max} + L_{core}, \tag{56}$$

where D_{max} is the maximum dynamic energy consumption of the core, L_{core} is the leakage energy of the core which can be obtained by measuring core power when the core is in halt mode [162].

However, Li *et al.* improves over such basic core power model by considering different levels of core speed. In the idle speed model, the amount of power consumed by a core in one unit of time is denoted as,

$$P_{core} = \rho s^{\alpha} = \left(\frac{\lambda}{m}\right) R s^{(\alpha-1)}, \tag{57}$$

where power allocated for a processor running on speed s is s^{α}, λ is the task arrival rate, m is the number of cores, R is the average number of instructions to be executed, ρ is the core utilization. The power consumed (P) by server S can be denoted by,

$$P = m\rho s^{\alpha} = \lambda R s^{(\alpha-1)}, \tag{58}$$

where $m\rho = \lambda\bar{x}$ represents the average number of busy cores in S. Since processor core consumes some power P^* even if it is idling, the above equation is updated as follows,

$$P = m(\rho s^{\alpha} + P^*) = \lambda R s^{(\alpha-1)} + mP^*. \tag{59}$$

In the constant speed model the parameter ρ becomes 1 because all cores run at the speed s even if there is no task to run.

Hence, the power consumption in the constant speed model can be denoted as,

$$P = m(s^{\alpha} + P^{*}). \tag{60}$$

However, in the constant speed model the CPU speed (i.e., the execution strategy) is kept constant. In general idle-speed model is difficult to implement compared to constant speed model because most of the current processors do not support running different cores at different speeds.

While CPU core can be treated as the smallest power consumer in a multicore CPU, another approach for accounting for power consumption of a multicore CPU is by modeling the power consumption as a summation of power consumed by its threads. Although it cannot be counted as a queuing theory based power model, Shi *et al.* described the amount of power consumed by a workload W in a multicore computer system as [163],

$$P = (P_{idle} + CP_t)T, \tag{61}$$

where P_{idle} is the idle power consumption, C is the concurrency level of the workload, and P_t is the average power dissipation of a thread. T is the total time taken to complete the workload. The CP_t accounts for the total dynamic power dissipation by all the threads executed by the multicore processor.

The second approach for multicore CPU power consumption modeling is component wise breakdown (i.e., additive) which deep dives into lower level details of the processor. Note that the rest of the power models described in this subsection are ordered based on the complexity of the power model. One notable yet simple power models in this category was introduced by Basmadjian *et al*. This is a generic methodology to devise power consumption estimation models for multicore processors [159]. They stated that the previous methods for modeling power consumption of multicore processors are based on the assumption that the power consumption of multiple cores performing parallel computations is equal to the sum of the power of each of those active cores. However, they conjectured that such assumption leads to the lack of accuracy when applied to more recent processors such as quad-core. They also took into consideration the parameters such as power saving mechanisms and resource sharing when estimating the power consumption of multicore processors. This approach had an accuracy within maximum error of 5%. Their power model can be denoted as,

$$P_{proc} = P_{mc} + P_{dies} + P_{intd}, \tag{62}$$

where P_{mc}, P_{dies}, and P_{intd} denotes the power consumption of chip-level mandatory components, the constituent dies, and inter-die communication. They also described power models for each of these power components. It can be observed that the power consumed by chip-level mandatory components and inter-die communication is modeled using/extending the Equation (7) which indicates even higher level system components may be modeled by adapting such lower level power models.

Another method for component wise breakdown is dividing the power consumption into dynamic and static power where dynamic power is due to power dissipated by cores, on-chip caches, memory controller (i.e., memory access). Then the total CPU power can be modeled as [164],

$$P_{proc} = P_{core} + \sum_{i=1}^{3} g_i L_i + g_m M + P_{base}, \tag{63}$$

where P_{base} is the base/static package power consumption. The component $\sum_{i=1}^{3} g_i L_i + g_m M$ represents the power consumption due to cache and memory access. Here, L_i is access per second to level i cache, g_i is the cost of a level i cache access. P_{core} can be often represented as $P_{core} = Cf^3 + Df$, where C and D are some constants [164]. Note that this is analogous with the server power model described in Equation (20). Furthermore, due to the cache and memory access is proportional to the CPU frequency, the above power model can be further simplified as,

$$P_{proc} = F(f) = af^3 + bf + c, \tag{64}$$

where a, b, and c are constants. Constants a and b are application dependent because cache and memory behavior can be different across applications. bf corresponds to cores' leakage power and power consumption of cache and memory controller. af^3 represents the dynamic power of the cores while $c = P_{base}$ is the base CPU power.

In a slightly more complicated power model, Jiménez *et al.* characterized the thermal behavior and power usage of an IBM POWER6™-based system [165]. This power model consists of several components which can be shown as,

$$P = N_{actc} P_{actc} + \alpha K + \beta L + \gamma M + \sigma N, \tag{65}$$

where N_{actc} is the number of active cores, the incremental power consumption increase for a core that exists its sleep state is represented by P_{actc}. The power consumption due to activity in the chip is modeled by using Instructions Per Cycle (IPC) is represented as K and the amount of L1 load misses per cycle (L1LDMPC) is represented as L. Similarly the memory system contribution to the power consumption is modeled by using the number of L2 misses per cycle (L2LDMPC and L2STMPC) which are represented by M and N respectively. α, β, γ, and σ are regression parameters.

In another work Bertran *et al.* developed a bottom-up power model for a CPU [166]. In a bottom-up processor power model the overall power consumption of the processor is represented as the sum of the power consumption of different power components. The power components are generally associated with micro-architecture components which allow the users to derive the power breakdown across the components. They modeled power consumption at each component level to obtain the final bottom-up power model. Their model is defined as,

$$P_{cpu} = \sum_{k=1}^{N} P_{dyn_k} + \sum_{k=1}^{M} SQ_k + RM + P_{uncore}, \tag{66}$$

which comprises of the power consumption of each hardware thread enabled on the platform, the SMT effect (SMT_{effect} denoted by S) of the cores with SMT enabled ($SMT\_enabled_k$

denoted by Q_k), the CMP effect as a function of the number of cores enabled (CMP_{effect} denoted by R), and the uncore power consumption (P_{uncore}). The total number of threads is denoted by N while total number of cores is denoted by M. They used system performance counters to calculate the parameters such as S in their power model. Furthermore, Bertran *et al.* presented a Performance Monitoring Counter (PMC) based power models for power consumption estimation on multicore architectures [167]. Their power modeling technique provides per component level power consumption. In another work Bertran *et al.* did a comparison of various existing modeling methodologies in Top-down and Bottom-up fashion [168].

In another such examples Bertran *et al.* used an accumulative approach for modeling multicore CPU power consumption assuming each core behaves equally [149]. Since all the cores behave equally, the single core model described in Equation (47) can be extended as follows for multicore model,

$$P_{total} = \left(\sum_{j=1}^{j=n_{core}} \left(\sum_{i=1}^{i=m} A_{ij} P_i \right) + P_{static} \right), \tag{67}$$

where P_i of each component is same as in the single core CPU power model described previously (in Equation (47)). However, A_{ij} need to be modified to perform per core accounting.

Similarly core level power model can be implemented without considering the power consumption of individual components. For example, Fu *et al.* described the processor power consumption $P(k)$ of a multicore processor as [169],

$$P(k) = P_s + \sum_{i=1}^{n} x_i(k) \left[P^i_{ind} + P^i_d(k) \right], \tag{68}$$

where k is a control period, P_s represents the static power of all power consuming components (except the cores). x_i represents the state of core C_i. If core i is active, $x_i = 1$, otherwise $x_i = 0$. The active power which is dependent on the frequency of the core is represented as $P^i_d(k) = \alpha_i f_i(k)^{\beta_i}$, where both α_i and β_i are system dependent parameters. The definition of active power $P^i_d(k)$ can be linked with the dynamic power consumption of a digital circuit described in Equation (7) where $\alpha_i(k)^{\beta_i}$ corresponds to ACV^2 of Equation (7).

A more detailed power model compared to the above mentioned two works (in Equations (67) and (68)) was described by Qi *et al.* [170] which is shown in Equation (69). Their power model was for multicore BP-CMP (block-partitioned chip-multiprocessor) based computing systems. Their power model had an additional level of abstraction made at block level compared to the previously described multicore power models. In their work they considered a CMP with 2^k processing cores, where $k \geq 1$. They assumed the cores are homogeneous. Considering the fact that it is possible to provide different supply voltages for different regions on a chip using voltage island technique, the cores on one chip was partitioned into blocks. In this model (which is shown below),

$$P = P_s + \sum_{i=1}^{n_b} \left(x_i P^i_{ind} + \sum_{j=1}^{nc_i} y_{i,j} P^{i,j}_d \right), \tag{69}$$

the P_s denotes the static power from all the power components, while x_i represents the state of the block B_i. It sets $x_i = 1$ if any core on the block is active and the block is on, otherwise $x_i = 0$ and B_i is switched off. P^i_{ind} is the static power of core i and it does not depend on the supply voltage or frequency. $y_{i,j}$ represents the state of the j'th core on block B_i. The frequency dependent active power for the core is defined as $P^{i,j}_d = C_{ef} . f^m_i$, where both C_{ef} and m are system dependent parameters. All the cores on block B_i run at the same frequency f_i. However, it does not depend on any application inputs. Note that the three power models (in Equations (67)–(69)) described in this subsection share many similarities.

In another processor energy consumption modeling work which cannot be attributed to either queuing theory based or component wise break down models, Shao *et al.* developed an instruction-level energy model for Intel Xeon Phi processor which is the first commercial many core/multi-thread x86-based processor [171]. They developed an instruction-level energy model for Xeon Phi through the results obtained from an energy per instruction (E_{pi}) characterization made on Xeon Phi. Their model is expressed as,

$$E_{pi} = \frac{(p_1 - p_0)(c_1 - c_0)/f}{N}, \tag{70}$$

where N is the total number of dynamic instructions in the microbenchmark used during the energy consumption characterization. The power consumed by microbenchmark is calculated by subtracting the initial idle power (p_0) from the average dynamic power (p_1). The initial idle power includes power for fan, memory, operating system, and leakage. The values c_0 and c_1 correspond to the cycle before the microbenchmark starts and the cycle right after the microbenchmark finishes respectively. Therefore, $(c_1 - c_0)$ corresponds to the total number of cycles executed by the microbenchmark. The f corresponds to the frequency at which the dynamic power is sampled.

D. Power Consumption of GPUs

Graphical Processing Units (GPUs) are becoming a common component in data centers because many modern servers are equipped with *General Purpose Graphical Processing Units (GPGPUs)* to allow running massive hardware-supported concurrent systems [172]. Despite the importance of GPU's on data center operations the energy consumption measurement, modeling, and prediction research on GPUs is still in its infancy. In this section we structure the GPU power models as performance counter based power models and as additive power models. Furthermore, we categorize the additive power models as pure GPU based models and as GPU-CPU power models.

The first category of GPU power models described in this paper are performance counter based power models. One of the examples is the work by Song *et al.* which combined hardware performance counter data with machine learning and advanced analytics to model power-performance efficiency for modern GPU-based computers [59]. Their approach is a performance counter based technique which does not require detailed understanding of the underlying system. They pointed out deficiencies of regression based models [128] such as Multiple Linear

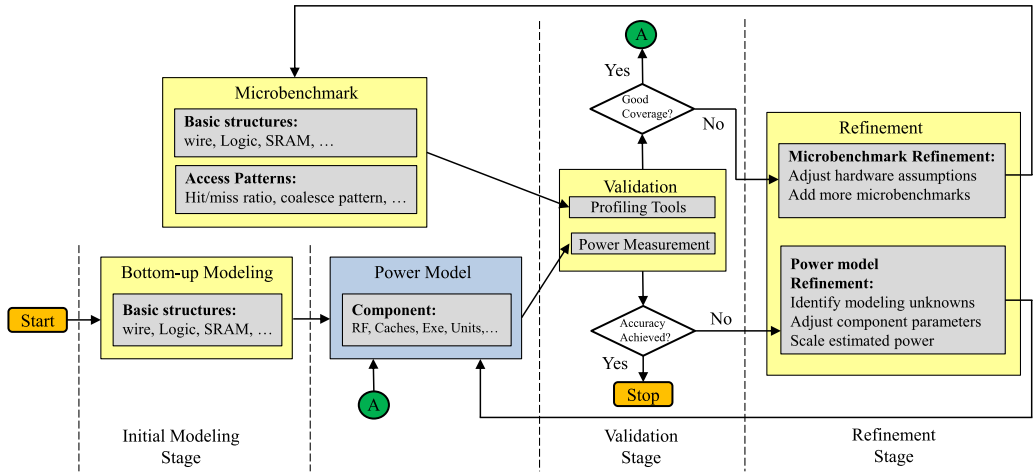


Fig. 12. GPUWattch methodology to build power models [175]. The process shown in this figure iteratively identifies and refines the inaccuracies in the power model.

Regression (MLR) for GPU power modeling [57] such as lack of flexibility and adaptivity.

In another work on use of GPU performance counters, sophisticated tree-based random forest methods were employed by Chen *et al.* to correlate and predict the power consumption using a set of performance variables [173], [174]. They showed that the statistical model predicts power consumption more accurately than the contemporary regression based techniques.

The second category of GPU power models are the additive power models. The first sub category is the pure GPU based additive power models. Multiple different software frameworks to model CPU/GPU power models currently exist which can be categorized as additive power models. In their work on GPUWattch, Leng *et al.* described a GPU power model that captures all aspects of GPU power at a very high level [175]. GPUWattch can be considered as an extension of Wattch CPU power model into the domain of GPUs. Fig. 12 shows an abstract view of the process followed in GPUWattch for building robust power models.

In this process a bottom-up methodology is followed to build an initial model. The simulated model is compared with the measured hardware power to identify any modeling inaccuracies by using a special suite of 80 microbenchmarks that are designed to create a system of linear equations that correspond to the total power consumption. Next, they progressively eliminate the inaccuracies by solving for the unknowns in the system. The power model built using this approach achieved an average accuracy that is within 9.9% error of the measured results for the GTX 480 GPU [175]. In a different work Leng *et al.* has also reported almost similar (10%) modeling error of GPUWattch [176]. The power model shown below is a very high level representation of the GPU power they model which consisted of the leakage($P_{leakage}$), idle SM (P_{idlesm}), and all components' (N in total) dynamic power as,

$$P = \sum_{1}^{N} (\alpha_i P_{max_i}) + P_{idlesm} + P_{leakage}, \tag{71}$$

where dynamic power of each component is calculated as the activity factor (α_i) multiplied by the component's peak power P_{max_i}. MCPAT is another framework similar to GPUWattch that models the power consumption of processors [143], [177]. In MCPAT the total GPU power consumption is expresses as a summation of the power consumed by different components of a GPU as [177],

$$\begin{aligned} P &= \sum P_{component}, \\ &= P_{fpu} + P_{alu} + P_{constmem} + P_{others}, \end{aligned} \tag{72}$$

where P_{fpu}, P_{alu}, and $P_{constmem}$ correspond to power dissipated by arithmetic and logic unit (ALU), floating point unit (FPU), and constant memory (Fig. 13).

Similar to the digital circuit level power breakdown shown in Equation (4), Kasichayanula *et al.* divided the total power consumption of a GPU Streaming Multiprocessor (SMs) into two parts called idle power and runtime power [178]. They modeled the runtime power (P) of the GPU as,

$$P = \sum_{i-1}^{e} (N_{sm} P_{u,i} U_{u,i}) + B_{u,i} U_{u,i}, \tag{73}$$

where N_{sm} is the number of components, $P_{u,i}$ is the power consumption of active component, e is the number of architectural component types, $B_{u,i}$ is the base power of the component, $U_{u,i}$ is the utilization. The block diagram of the McPAT framework.

It should be noted that the power consumption of GPUs change considerably due to the memory bandwidth utilization. See Fig. 14 for an example [179]. If the memory power can be reduced by half, it will lead to a 12.5% saving of the system power. From Fig. 14 it can be observed that the GPU cores (functional units) dominate the power consumption of the GPUs. Furthermore, the off-chip memory consumes a significant portion of power in a GPU.

In another example for additive power models, Hong *et al.* described modeling power consumption of GPUs [154]. They represented the GPU power consumption (P_{gpu}) as a sum of runtime power (*runtime_power*) and idle power (*IdlePower*). Runtime power is also divided among power consumption of Streaming Multiprocessors (SMs) and memory. To model the

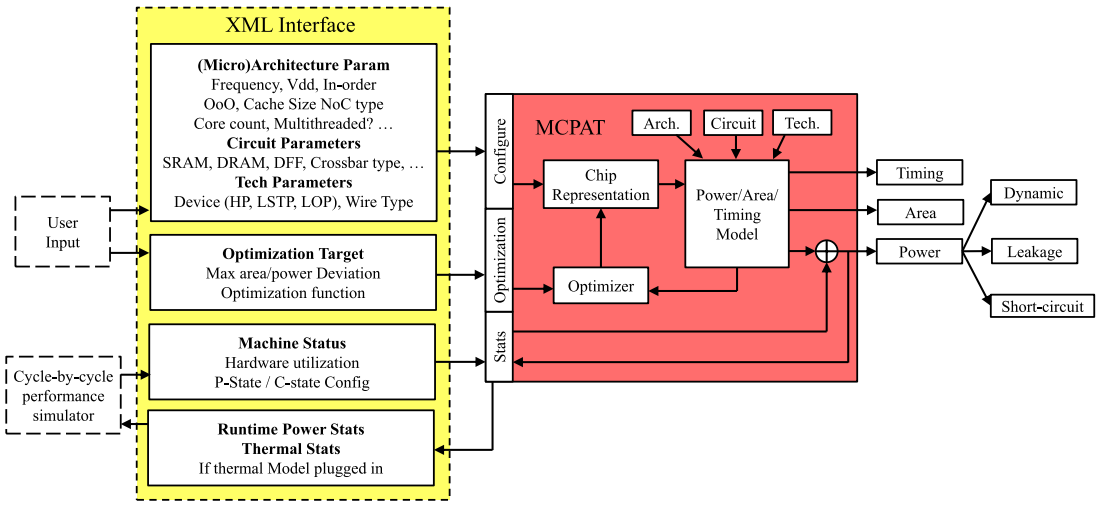


Fig. 13. Block diagram of the McPAT framework [143]. McPAT framework uses an XML-based interface with the performance simulator. McPAT is the first integrated power, area, and timing modeling framework for multithreaded and multicore/manycore processors [143].

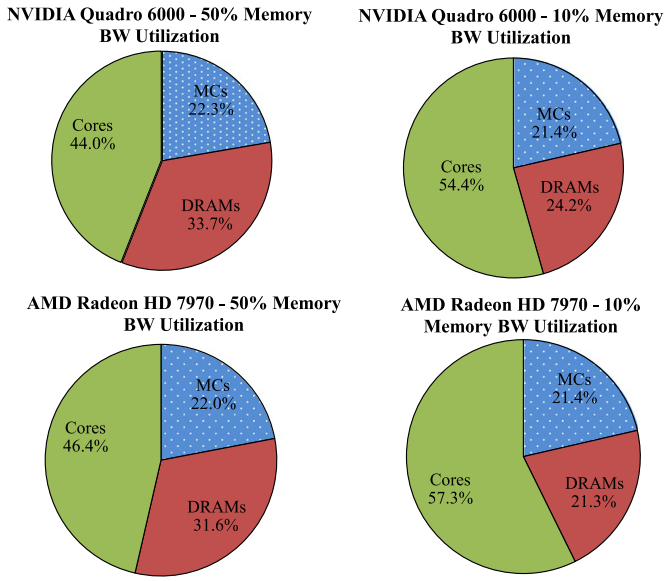


Fig. 14. Power breakdown for NVIDIA and AMD GPUs [179]. The off-chip DRAM accesses consume a significant portion of the total system power.

runtime power of SMs, they decomposed the SM into several physical components and accounted for the power consumption of each component. For example, *RP_Const_SM* is a constant runtime power component. Therefore, the complete GPU power model described by them can be denoted as,

$$\begin{aligned} P_{gpu} &= N_{sms} \sum_{i=0}^{n} S_i + P_{mem} + P_{idle}, \\ \sum_{i=0}^{n} S_i &= P_{int} + P_{fp} + P_{sfu} + P_{alu} + P_{texture} + P_{cc} \\ &\quad + P_{shared} + P_{reg} + P_{fds} + P_{csm}, \end{aligned} \tag{74}$$

where N_{sms} represents the number of streaming multiprocessors and a streaming component i is represented by S_i. Runtime power consumption of memory is represented by P_{mem} while the idle power is represented by P_{idle}. The terms P_{int}, P_{fp}, P_{sfu}, P_{alu}, $P_{texture}$, P_{cc}, P_{shared}, P_{reg}, P_{fds} correspond to integer arithmetic unit, floating point unit, SFU, ALU, Texture cache, Constant cache, shared memory, register file, FDS components of the GPU. P_{csm} is a constant runtime power component for each active streaming multiprocessor.

The second sub category of the additive GPU power models is the models that combine both the GPU and external (CPU) power consumption aspects. Certain works which model GPU power consumption as part of the full system power model currently exist. One such example is the power model described by Ren *et al.* considering the CPU, GPU, and main board components [180]. Their model can be represented as,

$$P_{system}(w) = \sum_{i=1}^{n} P^i_{gpu}(w^i) + \sum_{j=1}^{m} P^j_{cpu}(w^j) + P_{mainboard}(w), \tag{75}$$

where P_{system}, P_{cpu}, P_{gpu} and $P_{mainboard}$ represent the power of the overall system, GPU, CPU, and main board respectively. Number of GPUs and CPUs are represented as N and M which involve in the computing workload w. Workloads assigned to GPU_i and cpu_j are represented by w^i and w^j respectively. In a similar work [182] that expresses the power consumption of a GPU in the context of its corresponding CPU, the energy consumption of the GPU is expressed as,

$$E_{gpu} = t_{gpu}(P_{avg\_gpu} + P_{idle\_cpu}) + E_{transfer}, \tag{76}$$

where t_{gpu}, P_{avg_gpu}, P_{idle_cpu}, and $E_{transfer}$ represent the time spent on GPU, average power consumption of GPU, idle power consumption of CPU and the energy consumed for transfer between CPU and GPU respectively.

In a similar line of research Marowka *et al.* presented analytical models to analyze the different performance gains and energy consumption of various architectural design choices for hybrid CPU-GPU chips [181]. For such asymmetric CPU-GPU execution scenario, they assumed that a program's execution

TABLE IV
SUMMARY OF PROCESSOR POWER CONSUMPTION MODELING APPROACHES

Work(s)	Type	Characteristics	Limitations
[148][149]	single core/ multicore	Non-linear. Componentwise power breakdown (i.e., additive power models).	Each CPU subcomponent's power usage must be known beforehand.
[69][85] [156][158]	single core	Performance counter based	Choosing the proper performance counter(s) is a challenge, while certain performance counters may not be available across different processors.
[160][161]	multicore	Queuing theory based power models.	Power consumption of each core must be known beforehand.
[165]	multicore	Specific to IBM POWER6™	Need to know α, β, and σ variables beforehand.
[149][166]	multicore	Considers power consumption of each core and their subcomponents.	Need to know the subcomponents' power consumption beforehand.
[166][169][170]	multicore	Non-linear. Accounts for each core.	Power consumption of each core must be known beforehand.
[171]	multicore	Instruction-level energy consumption model made for Intel Xeon Phi	Need to know the initial idle power beforehand.
[59][173][174] [175]	GPU	Performance counter based power consumption models.	Based on multiple assumptions.
[154][178][179]	GPU	Component-wise breakdown of GPU power (i.e., additive power models).	Power consumption of each subcomponent must be known beforehand.
[178]	GPU	Non-linear. It is almost similar to a multicore power consumption model.	Power consumption of each GPU core must be known beforehand.
[180][181]	CPU-GPU (and other peripherals)	Non-linear power model.	Based on multiple assumptions.

time can be composed of a time fraction where the program runs sequentially $(1-f)$, and a time fraction of the program's parallel execution time where the program runs in parallel on the CPU cores (α), and a time fraction of the program's parallel execution time where the program runs in parallel on the GPU cores $(1-\alpha)$. They also assumed that one core is active during the sequential computation and consumes a power of 1, while the remaining $(c-1)$ idle CPU-cores consume $(c-1)k_c$ and g idle GPU-cores consume gw_gk_g. Therefore, during the parallel computation on the CPU-cores only, c CPU-cores consume c power and g idle GPU-cores consume gw_gk_g power. During the parallel computation on the GPU-cores only, g GPU-cores consume gw_g power and c idle CPU-cores consume ck_c power. In such scenario, the power consumption during the sequential, CPU, and GPU processing phases can be represented as,

$$P_s = (1-f)\left\{1 + (c-1)k_c + gw_gk_g\right\},$$
$$P_c = \frac{\alpha f}{c}\{c + gw_gk_g\},$$
$$P_g = \frac{(1-\alpha)f}{g\beta}\{gw_g + ck_c\}. \quad (77)$$

This requires $(1-f)$ to perform sequential computation, and times $\frac{(1-\alpha)f}{g\beta}$ and $\frac{\alpha f}{c}$ to perform the parallel computations on the GPU and CPU respectively. Note that $(1-f)$ is a time fraction of the program's parallel execution time where the program runs in parallel on the CPU cores (α), and $(1-\alpha)$ is a time fraction of the program's parallel execution time where the program runs in parallel on the CPU cores. Therefore, they represented the average power consumption W_a of an asymmetric processor as,

$$W_a = \frac{P_s + P_c + P_g}{(1-f) + \frac{\alpha f}{c} + \frac{(1-\alpha)f}{g\beta}}. \quad (78)$$

We list down a summary of the processor power consumption modeling approaches in Table IV.

VII. MEMORY AND STORAGE POWER MODELS

Traditionally, processors have been regarded as the main contributors to server power consumption. However, in recent times contribution made by memory and secondary storage for data center power consumption has increased considerably. In this section we first investigate on the memory power consumption and then move on to describing the power consumption of secondary storage such as hard disk drive (HDD) and flash based storage (SSD).

A. Memory Power Models

The second largest power consumer in a server is its memory [139]. Even in large petascale systems main memory consumes about ≈30% of the total power [183]. IT equipment such as servers comprise of a memory hierarchy. The rapid increase of the DRAM capacity and bandwidth has contributed for DRAM memory sub system to consume a significant portion of the total system power [184]. The DDR3 and Fully Buffered DIMM (FB-DIMM) dual in-line memory modules (DIMMs) typically consume power from 5W up to 21W per DIMM [139]. In the current generation server systems, DRAM power consumption can be comparable to that of processors [184]. Fig. 15 shows the organization of a typical DRAM [185]. Therefore, memory subsystem's power consumption should be modeled considering different components that make up the memory hierarchy. For example, Deng *et al.* divided power consumption of memory subsystem into three categories: DRAM, register/phase locked loop (PLL), and memory controller power [133]. However, most of the current studies on memory subsystem power consumption are based on Dynamic Random-Access

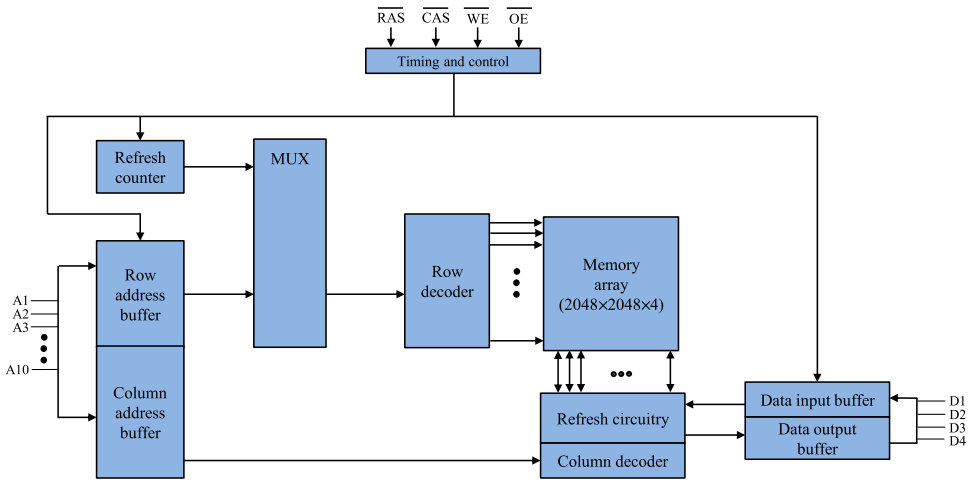


Fig. 15. Typical 16 Megabit DRAM ($4M \times 4$) [185]. Note that the DRAM includes a refresh circuitry. Periodic refreshing requires disabling access to the DRAM while the data cells are refreshed.

Memory (DRAM) power consumption. Commodity DRAM devices recently have begun to address power concerns as low power DRAM devices.

In the rest of this subsection we categorize the memory power models as additive power models and performance counter based power models. Furthermore, we organize the content from the least complex power models to most complex power models.

In the context of additive power modeling, one of the fundamental approaches is representing the power consumption as static and dynamic power. This can be observed in the context of DRAM power models as well. Lin *et al.* employed a simple power model in their work [186]. The model estimated the DRAM power (P_{dm}) at a given moment as follows,

$$P_{dm} = P_{static\_dm} + \alpha_1 \mu_{read} + \alpha_2 \mu_{write}, \tag{79}$$

where α is a constant that depends on the processor. The static power consumption of the DRAM is denoted by P_{static_dm}, while the read and write throughput values are denoted by μ_{read} and μ_{write} respectively.

In another additive power model, over all energy usage of the memory system is modeled as [187],

$$E = E_{Icache} + E_{Dcache} + E_{Buses} + E_{Pads} + E_{MM}, \tag{80}$$

where the energy consumed by the instruction cache, data cache is denoted by *Icache* and *Dcache* respectively. E_{Buses} represents the energy consumed in the address and data buses between *Icache*/*Dcache* and the data path. E_{Pads} denotes the energy consumption of I/O pads and the external buses to the main memory from the caches. E_{MM} is calculated based on the memory energy consumption model described in [155].

Another similar power modeling work was conducted by Ahn *et al.* [188]. Static power mainly comprises of power consumed from peripheral circuits (i.e., DLL and I/O buffers) such as transistors, and refresh operations. Since DRAM access is a two step process, dynamic power can be further categorized into two parts. First is the *activate precharge power* that is discharged when bitlines in a bank of a DRAM chip are precharged (During this process data in a row of the bank is delivered to the bitlines and latched (activated) to sense amplifiers by row-level commands). The second type of power is *read-write power* which is consumed when a part of the row is read or updated by column-level commands. The dynamic power consumption is proportional to the rate of each operation. Since a row can be read or written multiple times when it is activated, the rates of activate-precharge and read-write operations can be different. They modeled the total power consumed by a memory channel (i.e., total main memory power P_{mem}) as,

$$P_{mem} = DSR\sigma + E_{rw}\rho_{rw} + DE_{ap}f_{ap}, \tag{81}$$

where D is the number of DRAM chips per subset, S is the number of subsets per rank, R is the number of ranks per channel, σ is the static power of the DRAM chip, E_{rw} is the energy needed to read or write a bit, ρ_{rw} is the read-write bandwidth per memory channel (measured, not peak), E_{ap} is the energy to activate and precharge a row in a DRAM chip, and f_{ap} is the frequency of the activate-precharge operation pairs in the memory channel.

In an expanded version of the power models shown in Equations (79) and (81), Rhu *et al.* [189] modeled DRAM power as,

$$P = P_{pre\_stby} + P_{act\_stby} + P_{ref} + P_{act\_pre} + P_{rd\_bank} + P_{rd\_io} + P_{wr\_bank} + P_{wr\_io}, \tag{82}$$

where P_{pre_stby}, P_{act_stby},P_{ref}, P_{act_pre},P_{rd_bank},P_{rd_io}, P_{wr_bank}, P_{wr_io} repesent precharge standby power, active standby power, refresh power, activation & precharge power, read power and write power categorized as power consumed by DRAM bank and IO pins respectively. This power model was based on Hynix GDDR5 specification. Another similar power model based on background power and operation power was described by David *et al.* in [190]. They applied an extended version of this memory power model for Dynamic Voltage Frequency scaling where the memory power consumption at voltage v and frequency f is given by,

$$P_{f,v} = P_f - P_f P_{vstep} N_f, \tag{83}$$

where they conservatively assumed the total power reduction per voltage step as 6% thus $P_{vstep} = 0.06$.

Malladi *et al.* created a model to represent the expectation for memory energy $\mathbb{E}[E]$. During this process first they modeled a stream of memory requests as a *Poisson process* where they assumed that the time between requests follow an exponential distribution and the inter-arrival times are statistically independent [191]. When T_i is an exponentially distributed random variable for the idle time between two memory requests, the exponential distribution is parametrized by $1/T_a$ where T_a is the average inter-arrival time. They represented the power dissipation during *powerdown* and *powerup* as P_d and P_u respectively. If the idleness exceeds a threshold T_t the memory power-down happens, and a latency of T_u has to be faced when powering-up the memory. Powerdown is invoked with probability $f = P(T_i > T_t) = e^{-T_t/T_a}$ where f is the power down fraction. In this scenario the power dissipation by DRAM is P_d for $(T_i - T_t)$ time while powered-down and dissipates P_u for $(T_t + T_u)$ time while powered-up. T_i is the only random variable; $\mathbb{E}[T_i] = T_a$,

$$\begin{aligned}\mathbb{E}[E] &= f \times \mathbb{E}\left[P_d(T_i - T_t) + P_u T_t + P_u T_u\right] + (1-f)\mathbb{E}[P_u T_i] \\ &= f\left[P_d(T_a - T_t) + P_u T_t + P_u T_u\right] + (1-f)[P_u T_a] \\ &= P_d\left[f(T_a - T_t)\right] + P_u\left[f(T_t + T_u) + (1-f)T_a\right], \end{aligned} \quad (84)$$

where the expectation for memory energy is given by $\mathbb{E}[E]$.

Another power model for DRAM energy consumption was introduced by Lewis *et al.* in [92]. The model is based on the observation that energy consumed by the DRAM bank is directly related to the number of DRAM read/write operations involved during the time interval of interest. Energy consumption of a DRAM module over the time interval between t_1 and t_2 is expressed as,

$$E_{mem} = \int_{t_1}^{t_2} \left(\left(\sum_{i=1}^{N} C_i(t) + D(t) \right) P_{dr} + P_{ab} \right) dt, \quad (85)$$

where $C_i(t), i = 1, 2, \ldots, N$ is the last-level cache misses for all N constituent cores of the server when executing jobs, $D(t)$ is the data amount due to disk access or OS support and due to performance improvement for peripheral devices. P_{dr} is the DRAM read/write power per unit data. P_{ab} represents the activation power and DRAM background power. The value of P_{ab} was calculated by using the values mentioned in the DRAM documentation. In the case of AMD Opteron server used by Lewis *et al.*, the value of P_{ab} was amounted to 493 mW for one DRAM module.

All the above mentioned power models are additive power models. However, some power models are specifically developed using performance counter values (still these models can be considered as additive power models). In one such works Contreras *et al.* parametrized memory power consumption (P_{mem}) using instruction fetch miss and data dependencies [156],

$$P_{mem} = \alpha_1(B_{fm}) + \alpha_2(B_{dd}) + K_{mem}. \quad (86)$$

Here, α_1 and α_2 are linear "weighting" parameters. An important point about this power model is that it re-uses performance events used for CPU power model described in Equation (51). B_{fm} and B_{dd} correspond to instruction fetch miss and number of data dependencies respectively.

A completely different approach for modeling the memory power was followed by Roy *et al.* [85]. In their work on energy consumption of an algorithm A they modeled the memory power consumption as,

$$\begin{aligned} E_{mem}(A) = {} & P_{cke} T(A) + P_{stby} T_{act}(A)\alpha(A) \\ & + E_{act}\alpha(A) + (R(A) + W(A))\, T_{rdwr} P_{rdwr}, \end{aligned} \quad (87)$$

where $\alpha(A), R(A)$, and $W(A)$ represent the number of activation cycles (α and β pair), the number of reads, and writes respectively executed by A. $T_{act}(A)$ denotes the average time taken by one activation by A. The power model comprised of three components, the first component P_{cke} captures the leakage power drawn when the memory is in standby mode, with none of the banks are activated. The second component P_{stby} captures the incremental cost over and above the leakage power for banks to be activated and waiting for commands. The third component captures the incremental cost of various commands. Since α and β commands are always paired together, the energy cost of these two commands is represented as E_{act}. The energy usage of R and W commands is captured as $P_{rdwr} T_{rdwr}$.

B. Hard Disk Power Models

Hard Disk Drive (HDD) is currently the main type of secondary storage media used in data center servers. HDD contains disk platters on a rotating spindle and read-write heads floating above the platters. Disk is the subsystem that is hardest to model correctly [88]. This is because of the difficulty arising due to the lack of visibility into the power states of the hard disk drive and the impact of disk hardware caches.

Two components in the power consumed by HDDs (Disk drives in general) are called static power and dynamic power [193]. There are three sources of power consumption within a HDD: The Spindle Motor (SPM), Voice Coil Motor (VCM), and the electronics [194] (See Fig. 16 [192], [195]). The power consumption of electronics can be modeled by following the same techniques discussed under the section on CPU, and memory power consumption modeling. However, in the context of HDD the electromechanical components such as SPM accounts for most of the power consumption.

In this subsection we organize the HDD power models following a chronological ordering. One of the earliest work in this category is the work by Sato *et al.* [196]. The Power consumption by SPM can be expressed as [194], [196],

$$P_{spm} \approx n\omega_{spm}^{2.8}(2r)^{4.6}, \quad (88)$$

where, n is the number of platters of the HDD, ω_{spm} is the angular velocity of the SPM (i.e., RPM of the disk), and r is the radius of the platters. Since the platters are always rotating when the disk is powered, the above equation denotes the static power consumption by the disk irrespective of whether

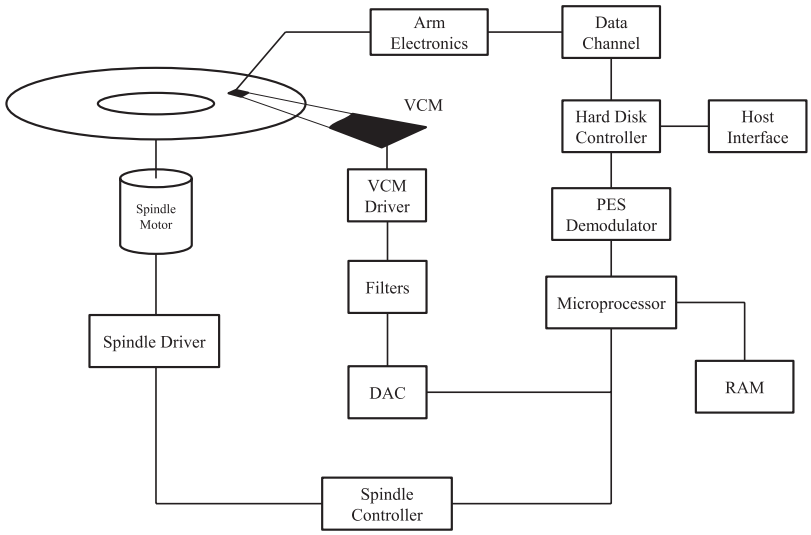


Fig. 16. Block diagram of a hard disk system [192]. The electromechanical subsystem consumes 50% of the total idle power with the remaining 50% dissipated in the electronics.

it is merely idling or actively performing I/O operations. The VCM power belongs to the dynamic portion of the HDD power consumption. VCM power consumption happens only when a disk seek needs to be performed. This also happens only during specific phases of a seek operation.

Hylick *et al.* observed that read energy consumption of multiple hard disk drives has a cubic relationship with Logical Block Number (LBN) [197]. Note that the amount of LBNs in a hard disk indicates its capacity. The read energy (E_r) consumption with the LBN (L) can be indicated as,

$$E_r \propto L^3. \tag{89}$$

Furthermore, they modeled the total amount of energy consumed by a drive servicing a set of N requests (considering I time idle) comprised of S seeks as [198],

$$E_{total} = \sum_{i=0}^{N} E_{active} + \sum_{i=0}^{S} E_{seek} + \sum_{i=0}^{I} E_{idle}, \tag{90}$$

where E_{total} is the total energy, E_{seek} is the seek energy, and E_{idle} is the idle energy in Joules.

Dempsey is a disk simulation environment which includes support for modeling disk power consumption [199]. To obtain a measure of the average power consumed by a specific disk stage S, Dempsey executes two workload traces which differ only in the amount of time spent in stage S. Then the average power consumption for disk stage S is represented using the following equation,

$$\bar{P}_s = \frac{E_2 - E_1}{T_2 - T_1}, \tag{91}$$

where E_i represents the total energy consumed by trace i and T_i is the total time taken by trace i. They referred this method of estimating the average power consumption of an individual stage as the *Two-Trace method*.

Hibernator is a disk array energy management system developed by Zhu *et al.* (circa year 2005) that provides energy savings while meeting performance goals [200]. They assumed that the most recent observed average request arrival rate at disk i in the disk array as α_i. For a disk that is spinning at speed j, the service time t_{ij} can be measured at run time. If the mean and the variance of the service time can be denoted as $K(t_{ij})$, the disk utilization ρ_{ij} can be calculated as $\rho_{ij} = \alpha_i K(t_{ij})$. If the disk i needs to change its speed in the new epoch, the disk cannot service requests while it is changing its spin speed. If the length of the transition period is denoted as T_i, if disk i does not service requests, the total energy for disk can be denoted as,

$$E_{ij} = P'_{ij} T_{epoch} \rho_{ij} + P''_{ij}(T_{epoch} - T_{epoch}\rho_{ij} - T_i) + P'''_{ij} T_i, \tag{92}$$

where E_{ij} is the energy consumption of keeping disk i at speed j in the next epoch. P'_{ij}, P''_{ij}, and P'''_{ij} correspond to active power, idle power at speed j, and transition power. The active time during which disk i is serving requests is $T_{epoch} \times \rho_{ij}$ since the request arrival rate is independent of the disk speed transition. The power components such as P'''_{ij} are simple divisions of the entire power at different states.

Bircher *et al.* estimated the dynamic events of the hard disk drive through the events such as interrupts and DMA access [150]. In another power model, the energy consumed by the VCM for one seek operation can be denoted as [193]–[195],

$$E_{vcm} = \frac{n J_{vcm} \omega_{vcm}^2}{2} + \frac{n b_{vcm} \omega_{vcm}}{3}, \tag{93}$$

where J_{vcm} is the inertia of the arm actuator, b_{vcm} is the friction coefficient of the arm actuator, and ω_{vcm} is the maximum angular velocity of the VCM [194]. When the average seek time t_{seek} is expressed as $t_{seek} = 2\frac{D_{avg}}{\omega_{VCM}}$ the power consumption of VCM can be modeled as,

$$P_{VCM} = \frac{E_{VCM}}{t_{seek}}, \tag{94}$$

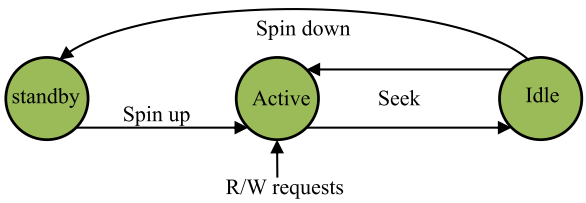


Fig. 17. Power state machine of a hard disk drive [202]. There are at least three power states in modern disk drives called active, idle, and standby.

where D_{avg} is the average angular seek distance. The power models for spindle motor (Equation (88)) and voice coil motor (Equation (94)) can be combined as follows to model the entire hard disk power consumption [195],

$$E = P_{SPM}t_0 + P_{VCM}t_2 + E_c, \tag{95}$$

where $t2$ is the actual seek time while E_c correspond to the energy consumption of the electronic part of the disk system ($E_c \approx 40\%$ of total system idle power).

Similar to other components of a server, the HDD power consumption is dependent on low level issues such as the physical design of the drive, dynamic behavior of the electromechanical components [194]. However, most of the power modeling techniques represent disk power consumption at higher level abstraction based on HDD power states. Each state corresponds to a category of high level activity and each state has an associated cost. Transitioning between different states may also incur power costs and extra latencies. The power consumption of a HDD can be shown as a state machine where the nodes correspond to the power states and the edges denote the state transitions (See Fig. 17).

At least three modes of operation (i.e., power states) exists for power-manageable disk: active, idle, and standby (See Fig. 17) [201], [202]. In the idle state the disk continues spinning, consuming power P_{id}. In the active state the disk consumes power P_{act} for transferring data while in the standby state, the disk is spun down to reduce its power consumption to P_{sb}. The remaining time is the idle time when disk i is spinning at speed j, but does not service requests.

In another HDD power modeling attempt, Bostoen *et al.* [201] modeled the energy dissipation in the idle mode (E_{id}) as a function of the idle time T_{id} (the time between two I/O requests) as follows,

$$E_{id}(T_{id}) = P_{id}T_{id}, \tag{96}$$

while in the standby mode, the disk energy consumption E_{sb} can be represented as a function of idle time T_{id} as,

$$E_{sb}(T_{id}) = P_{sb}(T_{id} - T_{du}) + E_{du} = P_{sb}T_{id} + E'_{du}, \tag{97}$$

where $E'_{du} = E_{du} - P_{sb}T_{du}$ is the extra energy dissipated during disk spin-down and spin-up taking the standby mode as a reference.

A summary of the HDD power models is shown in Table V.

C. Solid-State Disk Power Models

Flash memory based solid state disks (also known as solid state drives) (SSD) are becoming a crucial component of modern data center memory hierarchies [203], [204]. SSD has become a strong candidate for the primary storage due to its better energy efficiency and faster random access. Design of the flash memories is closely related to the power budget within which they are allowed to operate. Flash used in consumer electronic devices has a significantly lower power budget compared to that of an SSD used in data centers.

In this subsection we organize the power models based on their chronological order. In a typical Flash memory device, multiple silicon Flash memory dies are packed in a 3D stacking. These Flash memory dies share the I/O signals of device in a time-multiplexed way. Park *et al.* termed the I/O signals of the package as *channel* while they termed a memory die as *way*. Park *et al.* expressed the per-way power consumption of write operation (P_{pw}) [205] as,

$$P_{pw} = \frac{(P_{sw} - P_{idle})}{\#\text{ active Flash dies at the plateau}}, \tag{98}$$

where the denominator of the right-hand side represents the number of flash dies that are turned on during the plateau period of the write operation. The power consumption by sequential write operations is denoted by P_{sw}. In their work they measured the power during the plateau period where all the flash dies and the controller are active. Per-way power consumption of read operation can be calculated in the same manner.

Among the multiple different power consumption models for SSDs, Mohan *et al.* presented *FlashPower* which is a detailed power model for the two most popular variants of NAND flash [206] called single-level cell (SLC) and 2-bit multilevel cell (MLC) based NAND flash memory chips. FlashPower uses analytical models to estimate NAND flash memory chip energy dissipation during basic flash operations such as *read*, *program* and *erase*, and when the chip is *idle*. While they modeled the Flash power consumption in a very detailed manner we highlight three of the highest level models they described in their work below. They modeled the total energy dissipated per read operation for SLC flash and fast page of MLC flash as,

$$E_r = E_{sp,r} + E_{up,r} + E_{bl1,r} + E_{bl0,r} + E_{sl,r} + E_{rpre} + E_{ss,r} + E_{dec,r}, \tag{99}$$

where $E_{sp,r}$ is the energy to bias the wordline of the selected page to ground, $E_{up,r}$ is the energy to bias the unselected pages to V_{read}, E_{rpre} is the energy to transit from the read to the precharge state (E_{rpre}), $E_{dec,r}$ is the energy for the decode operation estimated using CACTI [207], $E_{bl1,r}$ and $E_{bl0,r}$ correspond to the energy dissipated during transitioning from logical "1" to "0" and vice versa. They used the CACTI's DRAM sense amplifier model to find the amount of energy dissipation for sensing. The term $E_{ss,r}$ corresponds to this term.

In a similar energy modeling attempt, Mohan *et al.* denoted the total energy dissipated when programming the SSD (i.e., program operation) as,

$$E_p = E_{dec,p} + E_{pgm} + E_{ppre}, \tag{100}$$

TABLE V
SUMMARY OF MEMORY POWER CONSUMPTION MODELING APPROACHES

Work(s)	Type	Characteristics	Limitations
[186]	DRAM	Additive power model. Considers static, read, and write power of the entire DRAM	α is dependent on the system's processor.
[188]	DRAM	Additive power models. Multiple DRAM chips. More elaborated one compared to [186].	The frequency of activate-precharge operation pairs depends on many factors.
[189]	DRAM	Additive power model which is more descriptive than [188]. Considers multiple subcomponents of DRAM power dissipation.	The values of the individual power dissipation need to be identified beforehand.
[191]	DRAM	Additive power model. Based on information theory.	Assumes that power up latency is exposed on the critical path.
[92]	DRAM	Performance counter based. Considers multiple factors such as disk accesses, peripheral devices, etc.	The traffic over hyper transfer buses affect the value of [92].
[85]	DRAM	Performance counter based. Energy consumption during an algorithm's execution.	Considers only a limited set of algorithms.
[156]	DRAM	Performance counter based.	The values of the weighting parameters α_1 and α_2 need to be known beforehand.

where $E_{dec,p} = E_{dec,r}$ which is estimated using CACTI, E_{pgm} is the maximum energy for programming, E_{ppre} is the energy to transit from program state to precharge state. In the erasure operation of a flash memory, the erasure happens at the block level. The controller sends the address of the block to be erased. The controller uses only the block decoder and the energy for block decoding ($E_{dec,e}$) is calculated using CACTI. They modeled the total energy dissipated in the erase operation E_{erase} as,

$$E_{erase} = E_{dec,e} + \sum_{i=0}^{N_{ec}} E_{se}(V_{se,i}) + E_{epre}, \qquad (101)$$

where $E_{se}(V_{se,i})$ is the energy for suberase operation with $V_{se,i}$ is the voltage used for that purpose where i denotes the iteration count of the suberase operation. The erase operation ends with a read operation. They took the energy for transitioning from the erase to precharge as the same as energy to transition from read to precharge ($E_{epre} = E_{rpre}$).

The power state transition between these different states used by FlashPower is shown in Fig. 18. Note that Fig. 18(a) and (b) show the power transition for SLC and MLC NAND flash chips respectively. The bubbles show the individual states while solid lines denote state transitions. A summary of the SSD power models is shown in Table VI.

D. Modeling Energy Consumption of Storage Servers

Storage systems deployed in data centers account for considerable amount of energy (ranked second in the energy consumption hierarchy as described in Section I) consumed by the entire data center. Some studies have shown that the energy consumption of storage may go up to 40% of the entire data center [208]. The storage portion of a data center consists of storage controllers and directly-attached storage [209]. Power consumption of storage systems is unique because they contain large amounts of data (often keeps several copies of data in higher storage tiers). In backup storage systems, most of the data is cold because backups are generally only accessed when there is a failure in higher storage tier [208].

Inoue *et al.* conducted power consumption modeling of a storage server [210]. They introduced a simple power consumption model for conducting storage type application processes where the maximum electric power of a computer is consumed if at least one storage application process is performed on the computer. They used a power meter to measure the entire storage server's power since it is difficult to measure the power consumption at individual component level. They measured the power consumption rate of the computer in the environment $\phi(m, w)$ where $w(0 \leq w \leq 1)$ is the ratio of W processes to the total number m of concurrent processes. They measured the power consumption rate $e_\phi(t)$ where ten processes are concurrently performed ($m = 10$). Their read (R) Write (W) energy consumption model can be represented as,

$$e_\phi(t) = \begin{cases} (a)\ \max\ W, & \text{if } m \geq 1 \text{ and } w = 1, \\ (b)\ \max\ R, & \text{if } m \geq 1 \text{ and } w = 0, \\ (c)\ \frac{(12.66w^3 - 17.89w^2 + 9.11w)(\max\ W - \max\ R)}{3.88 + \max\ R}, & \\ \text{if } m\ \geq 1 \text{ and } 0 < w < 1, & \\ (d)\ \min\ E \quad \textit{if } m = 0, & \end{cases} \qquad (102)$$

where max W and max R correspond to the maximum rate at which the read and write operations are performed. They did experiments in a real environment and obtained an equation (Equation (102) (c)) for power consumption when concurrent processes are performed. While we list down this storage server power model in this subsection, most of the data center storage power modeling attempts were described in Section VII-A–C.

VIII. DATA CENTERS LEVEL ENERGY CONSUMPTION MODELING

The power models described in this paper until this point have been focused on modeling energy consumption at individual components. When constructing higher level power models for data centers it is essential to have knowledge on the details of such lower level components which accounts for the total data center power consumption. This section investigates on the data center power models constructed on the higher levels of abstractions. First, we describe modeling the energy consumption of a group of servers and then move on to describing the efforts on energy consumption modeling of data center networks. Modeling the data center cooling power consumption

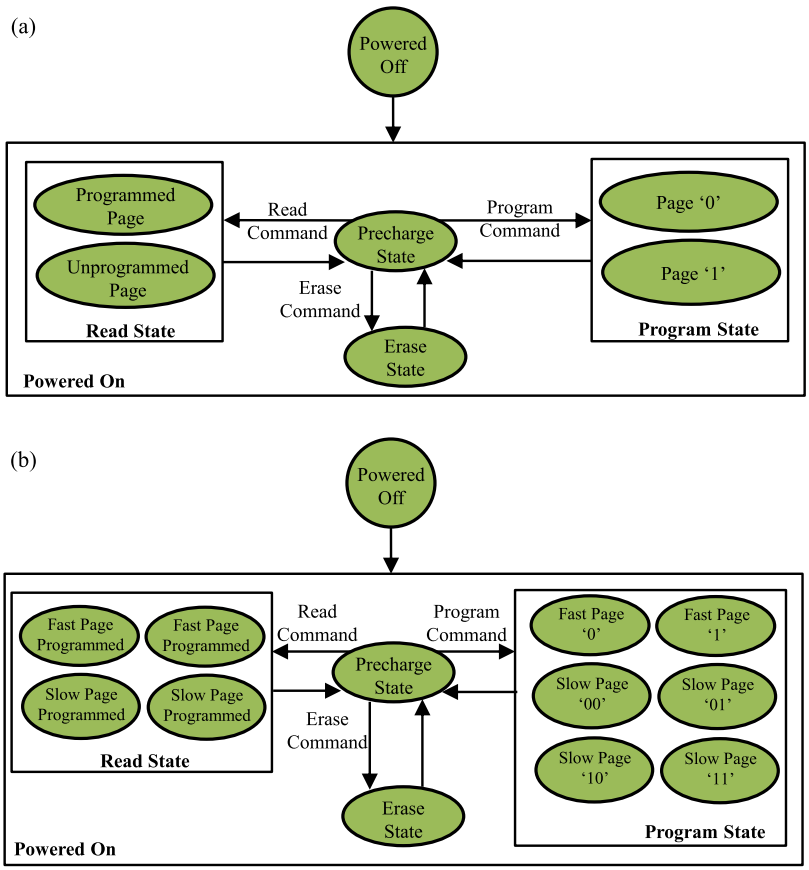


Fig. 18. Power state machines of NAND Flash [206]. (a) Power state machine for a SLC NAND Flash chip. (b) Power state machine for a 2-bit MLC NAND Flash chip.

TABLE VI
SUMMARY OF DISK POWER CONSUMPTION MODELING APPROACHES

Work(s)	Types	Characteristics	Limitations
[194][196]	SPM	Non-linear model. Accounts for the physical characteristics.	Simple power model.
[193][194][195]	VCM	Non-linear model. Accounts for the physical characteristics.	More complicated power model compared to [194][196]
[198]	Complete HDD	Accounts for three different states of operations.	Simple power model
[197]	HDD	Accounts only for the read energy	Simple power model
[200]	HDD	Considers the transitioning time periods of the disk.	More complicated power model.
[201]	HDD	Accounts for the energy dissipated during the idle mode.	Simple model, multiple assumptions.
[199]	HDD	This is a simulation based model developed using Dempsey.	Needs physical metering.
[206]	SSD	Describes the SSD power consumption in detail.	Depends on CACTI
[205]	SSD	Write power consumption of Flash memory.	The modeled targets the plateau of the Flash memory's operation.

distribution and the metrics for data center energy efficiency are described next.

A. Modeling Energy Consumption of a Group of Servers

The aforementioned lower level power models can be extended to build higher level power models. While the basics remains the same, these higher level power models pose increased complexity compared to their lower level counterparts. One of the frequently sought higher level abstractions is a group of servers. In this subsection we investigate on various different techniques that have been followed for modeling the power consumption of a group of servers. The power models developed for group of servers can be categorized into three subcategories as queuing theory based power models, power efficiency metrics based power models, and others.

First, we consider use of queuing theory for modeling energy consumption of a group of servers. Multiple different types of parameters need to be considered in the context of a group of servers compared to a single server's power model. For example, there is a time delay and sometimes a power penalty associated with the setup (turn the servers ON) cost. Gandhi *et al.*, Artalejo *et al.*, and Mazzucco *et al.* have studied about server farms with setup costs specifically in the context of modeling the power consumption of such systems [211]–[213] which we elaborate next.

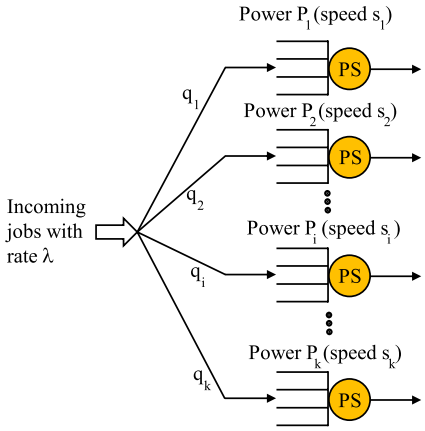


Fig. 19. K-server farm model [214]. The model assumes that the jobs at a server are scheduled using the Processor-Sharing (PS) scheduling discipline.

In their work Gandhi *et al.* developed a queuing model for a server farm with k-servers (see Fig. 19) [211], [214]. They assumed that there is a fixed power budget P that can be split among the k servers in the cluster, with allocating P_i power to server i where $\sum_{i=1}^{k} P_i = P$. In their work they modeled the server farm with setup costs using M/M/k queuing system, with a Poisson arrival process with rate λ with exponentially distributed job sizes, denoted by random variable $S \sim Exp(\mu)$. They denoted the system load as $\rho = \frac{\lambda}{\mu}$ where $0 \leq \rho \leq k$. For stability they need $\lambda < k\mu$. In their model a server can be in one of four states: on,idle,off, or setup. When a server is serving jobs its in the on state and its power consumption is denoted by P_{on}. If there are no jobs to serve, the server can either remain idle, or be turned off, where there is no time delay to turn off a server. Furthermore, if a server remains idle, it consumes non-zero power P_{idle} and they assumed $P_{idle} < P_{on}$. The server consumes zero power if it is turned off. Hence $0 = P_{off} < P_{idle} < P_{on}$. To transit a server from off to on mode it must under go setup mode where it cannot serve any requests. During the setup time, the server power consumption is taken as P_{on}. They denoted the mean power consumption during the ON/IDLE state as,

$$P_{on|idle} = \rho P_{on} + (k - \rho)P_{idle}, \tag{103}$$

where ρ is the expected number of on servers and $(k - \rho)$ is the expected number of idle servers.

A sightly different version of the above power model was described by Lent [215]. This model assumed that the servers are homogeneous and there exists an ideal load balancing among the nodes that are running. The power consumption of the computing cluster is modeled as,

$$P(\lambda) = nk(I + J\rho) + n(1 - k)H, \tag{104}$$

where λ is the job arrival rate. $k = m/n$ is the ratio of running to hibernating nodes. I is the idle power consumed by each server while j is the power increment for utilization level of ρ. H is the node power consumption while in hibernate mode. Note that the original power model described in [215] has an additional term called O which describes the power consumption of other equipment in the data center facility such as UPSs, network equipment, etc.

In [211] servers are powered up and down one at a time [216]. Mitrani *et al.*, worked on the problem of analyzing and optimizing the power management policies where servers are turned on and off as a group [216]. They mentioned that in a large-scale server farm it is neither desirable nor practical to micro-manage power consumption by turning isolated servers on and off. In their model the server farm consists of N servers of which m are kept as reserve ($0 \leq m \leq N$). The jobs arrive at the farm in a Poisson stream with a rate λ. The operational servers accept one job at a time and the service times are distributed exponentially with a mean of $1/\mu$. The availability of the reserve servers is controlled by two thresholds U and D ($0 \leq U \leq D$). If the number of jobs in the system increases from U to $U + 1$ and the reserves are OFF they are powered ON as a block. They become operational together after an interval of time distributed exponentially with mean $1/\nu$ in which the servers are consuming power without being operational. Even if $U \geq N$ by the time the reserves are powered on jobs may have departed leaving some or all of them idle. Similarly, if the reserves are powered ON (or powered up) and the number of jobs in the system drops from $D + 1$ to D, then they are powered down as a block. When all servers are ON, the reserves are not different from other servers, the system behaves like an M/M/N queue.

In a slightly different work from [211] where it does not allow for jobs to queue if no server is available, Mazzucco *et al.* modeled the power consumption by a cluster of n powered on servers as [213],

$$P = ne_1 + \bar{m}(e_2 - e_1), \tag{105}$$

where e_1 is the energy consumed per unit time by idle servers. Energy drawn by each busy server is denoted by e_2 while the average number of servers running jobs is denoted by $\bar{m}$ ($\bar{m} \leq n$, $\bar{m} = \left\lceil \frac{T}{\mu} \right\rceil$, where T is the system's throughput, $\frac{1}{\mu}$ is the service time). While Gandhi *et al.*, Mitrani *et al.*, and Mazzucco *et al.* described interesting observations related to the power consumption of server farms, such observations are out of the scope of this survey. More details of these observations are available from [211].

In another energy consumption modeling attempt for group of data center servers following a system utilization based model was created by Liu *et al.* [217]. Different from the power model in Equation (22), the Liu *et al.*'s work considers u as the average CPU utilization across all servers in the group of servers being considered. They calculate the IT resource demand of interactive workload i using the M/G1/1/PS model, which gives $\frac{1}{\mu_i - \lambda_i(t)/a_i(t)} \leq rt_i$. Their IT power model which covers all the IT energy consumption of a data center is defined as,

$$d(t) = \frac{\sum_i a_i(t)}{Q}(P_i + (P_b - P_i)u_i) + \frac{\sum_j n_j(t)}{Q}P_b, \tag{106}$$

where $u_i = \left(1 - \frac{1}{\mu_i rt_i} + \frac{\sum_j n_{ji}(t)}{a_i(t)}\right)$. This power model has been derived by using the server power model shown in Equation (22).

The term $a_i(t)$ represents the minimum CPU capacity needed, which can be expressed as a linear function of the arrival rate $\lambda_i(t)$ as $a_i(t) = \frac{\lambda_i(t)}{\mu_i - 1/rt_i}$. The value of μ_i is estimated through real measurements and the response time requirements rt_i based on the SLAs. Each batch job is denoted by j and at time t it shares $n_{ji}(t) \geq 0$ CPU resource with interactive workload i and uses additional $n_j(t) \geq 0$ resources by itself. Furthermore, they used an IT capacity constraint as $\sum_i a_i(t) + \sum_j n_j(t) \leq D$ assuming the data center has D CPU capacity in total.

The second clearly observable category of power models for group of servers is the power models developed based on a data center performance metric. Unlike the lower level power models, it can be observed that such power efficiency metrics are utilized to create power consumption models at the higher levels of data center power hierarchy. Use of PUE metric for calculating the non-IT power consumed by servers is a simple way to relate the Non-IT power with the servers. One such example is where Qureshi *et al.* [218] created a power consumption model for a group of n servers deployed in an Internet scale system by merging the power model described in Equation (22) with the *Power Usage Effectiveness (PUE)* metric of the data center as follows,

$$P \approx n\left(P_{idle} + (P_{peak} - P_{idle})U + (\eta_{pue} - 1)P_{peak}\right), \quad (107)$$

where n denotes the server count, P_{peak} is the server peak power in Watts, P_{idle} is the idle power, and U is the average server utilization, and PUE is denoted by η_{pue}. In this model $(\eta_{pue} - 1)$ term denotes the ratio between the non-IT power and IT-Power in the context of a server. Hence, the term $(\eta_{pue} - 1)P_{peak}$ corresponds to the amount of non-IT power being apportioned to a server. Overall this power model tries to unify the entire data center energy consumption under the unit of a server which distinguishes this power model from Fan *et al.*'s work in Equation (22).

A different perspective of modeling the power consumption by a group of servers was made by Pedram [219]. Total power consumption of a data center which consists of N server chassis was modeled by Pedram. The data center power model can be shown as,

$$P_{dc} = \left(1 + \frac{1}{\eta(T_s)}\right)\sum_{i=1}^{N} P_i^{ch}, \quad (108)$$

where η represents the *Coefficient of Performance (COP)* which is a term used to measure the efficiency of a CRAC unit. The COP is the ratio of heat removed (Q) to the amount of work necessary (W) to remove that heat [220]. P_i^{ch} denotes the total power consumption of a chassis. Note that a chassis may host multiple servers (each consuming power P_j^s) and the chassis may consume a baseline power P_i^b (which includes fan power and switching losses). Hence the total chassis power can be stated as,

$$P_i^{ch} = P_i^b + \sum_j P_j^s. \quad (109)$$

In the rest of this subsection we describe the power models of group of servers which cannot be categorized under the queuing theory based or energy efficiency metric based power models. In one such works a parallel system's power consumption was expressed as [56], [59], [221],

$$E = \sum_{i=1}^{\Omega}\int_{t_1}^{t_2} P_i(t)dt = \sum_{i=1}^{\Omega}\bar{P}_i T_{delay}, \quad (110)$$

where energy E specifies the total number of joules in the time interval (t_1, t_2), as a sum product of the average power $(\bar{P}_i)$ (i $\in$ set of all the nodes Ω) times the delay $(T_{delay} = t_2 - t_1)$ while power P_i (i $\in$ set of all the nodes Ω) describes the rate of energy consumption at a discrete point in time on node i. This is essentially an extension for the power model described in Equation (24) (However, in this case it is for entire system.). A similar extension of a single server power model to represent a server cluster power was made by Elnozahy *et al.* [108] where they represented the power consumption of a cluster of n identical servers operating at frequency f as [72],

$$P(f) = n \times (c_0 + c_1 f^3), \quad (111)$$

where all the parameters remain the same as described in Equation (20).

Aikebaier *et al.* modeled energy consumption of a group of computers [222] with two power consumption models: simple and multi-level models. They considered the scenario of a system S which is composed of computers $c_1, \ldots, c_n (n \geq 1)$ with $m \geq 1$ processes $p_1, \ldots, p_m$ running [223]. First they measured the amount of electric power consumed by web applications in a cluster system composed of Linux Virtual Server (LVS) systems. They abstract out the essential properties which dominate the power consumption of a server [224]. Based on the measurement results they presented models for energy consumption of a computer. They denoted $E_i(t)$ as the electric power consumption (Watts per time unit) of a computer c_i at time $t[W/\text{time unit}](i = 1, \ldots, n)$. Furthermore, $\max E$ and $\min E$ indicate the maximum energy consumption and minimum energy consumption of a computer c_i [129], respectively $(\min E_i \leq E_i(t) \leq \max E_i,\ \max E = \max(\max E_1, \ldots, \max E_n),\ \min E = \min(\min E_1, \ldots, \min E_n))$. In the simple model, the normalized energy consumption rate $e_i(t)$ is given depending on how many number of processes are performed as,

$$e_i(t) = \begin{cases} \max\ e_i, & \text{if} \quad N_i(t) \geq 1, \\ \min\ e_i, & \text{if} \quad N_i(t) < 1, \end{cases} \quad (112)$$

where $\min e_i$ and $\max e_i$ correspond to $\min E_i/\max E$ and $\max E_i/\max E$ respectively. This model simply says that the power consumption can vary between two levels $\min e_i$ and $\max e_i$ for computer c_i. In this model they assumed that the electric power is maximally consumed even if a single process is run by the computer. They developed a multi-level model extending the aforementioned power model as well [129].

While the above server group power models considered only power consumption by servers, there are certain other work

TABLE VII
SUMMARY OF POWER CONSUMPTION MODELING APPROACHES FOR A GROUP OF SERVERS

Work(s)	Characteristics	Limitations
[211][212][213]	Models server farms with setup costs.	Depends on multiple assumptions.
[56][59][221]	An extension of the power model 24.	Relatively simple power model.
[222][223][224]	Two power models: simple and multi-level. Both are based on process level power consumption.	Measuring the power consumed by each process is a challenge.
[218]	Hybrid of the power model described in Equation (22) and the PUE metric of the data center	Need to measure idle and peak powers of each server.
[217]	Derived using the server power model described in Equation (22)	Depends on multiple assumptions.
[219]	Considers chassis power consumption aspects.	Depends on the server mounting architecture.
[108]	Extension of a single server power model to represent a group of server's power usage.	Assumes all the servers are homogeneous.

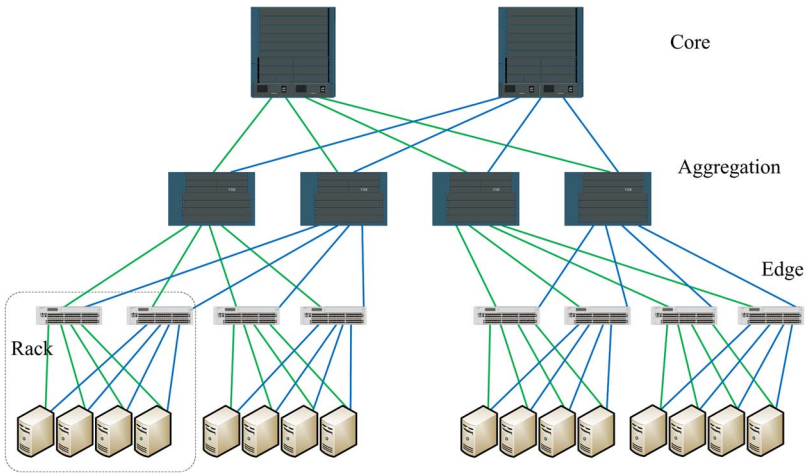


Fig. 20. A typical fat-tree data center network [227]. Many data centers are built using a fat-tree network topology due to its high bisection bandwidth [231].

that model the energy consumption of a group of servers along with the network devices which interconnect them. While such work can be listed under Section VIII-B or Section VIII-F, we list down such work here since they are more specific to the compute cluster. The work by Velasco *et al.* is an example for such power modeling attempt [225]. They modeled the power consumption of cluster i, $P^i_{cluster}$ as,

$$P^i_{cluster} = a^i \left(\frac{M}{2}(P_{agg} + P_{edge}) + \sum_{s=1}^{M^2/4} P_{server}\left(k^i_s\right) \right) \quad (113)$$

where a^i is a term which indicates whether the cluster is active and P_{agg} and P_{edge} denote the power consumption of aggregation and edge switches. When combined with a term for power consumption of core switches, the data center's total IT devices' power consumption can be modeled as,

$$P_{IT} = \frac{M^2}{4} P_{core} + \sum_{i=1}^{M} P^i_{cluster}. \quad (114)$$

It could be observed that in multiple server group level power models (or even at the server level power models such as shown in Equations (29), (30), and (25)) the homogeneity of the servers is assumed across the server cluster. While the energy consumption behavior might be similar across multiple servers, the models' constants might change slightly per server basis due to slight differences in electro-mechanical characteristics of the hardware components of the servers. A summary of power consumption modeling approaches for groups of servers is shown in Table VII.

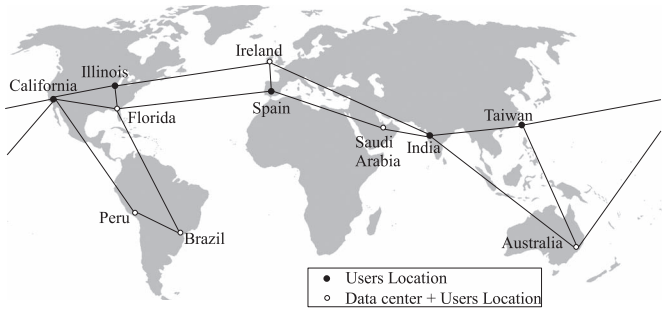


Fig. 21. A distributed data center network [225]. Each location collects user traffic towards the set of federated data centers which consists of five data centers strategically located around the globe.

B. Modeling Energy Consumption of Data Center Networks

In this section we discuss the power modeling work conducted on a group of servers connected via a data center network as well as the scenarios such as multiple data center systems being linked via wide area networks (i.e., distributed data centers [226], see Fig. 21 for an sample [225]). When modeling energy consumption at such higher level abstraction, we need to consider the energy cost of communication links and intermediate hardware (See Fig. 20 for an example of a data

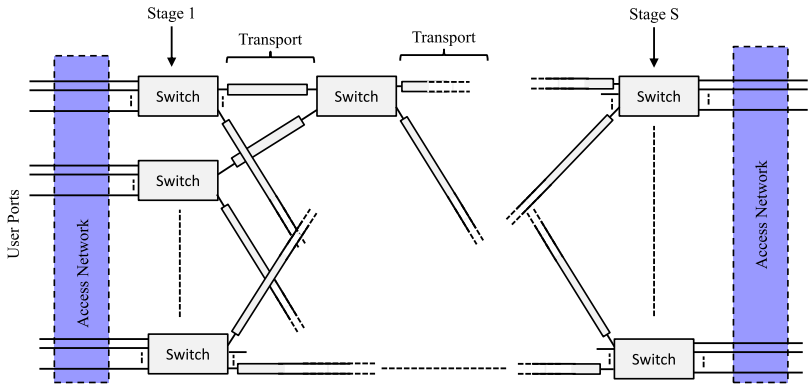


Fig. 22. Simplified model of a global network [233]. The network comprises of an array of multiport switches interconnected by optical transport systems and connected to user devices via an access network.

center network [227], [228]). Such energy costs are significant in the global Internet infrastructure. Most of the existing data center networks exhibit very small *dynamic range* because the idle power consumption is relatively significant compared to the power consumption when the network is fully utilized [229]. Multiple research have been conducted to address such network power consumption issues [230].

Next, we move on to exploring the data center network level energy consumption modeling efforts.

1) Modeling Network Link Power Consumption: Energy consumption modeling conducted focusing the data center networks consider the entire network's energy consumption including network devices as well as network links. We follow chronological ordering of power models in this subsection.

Additive power consumption models are present at the data center network level. Heller *et al.* described the total network energy consumption based on the level of power consumption by a link. Their energy model can be expressed as [227],

$$E_{net} = \sum_{(u,v)\in E} X_{u,v}a(u,v) + \sum_{u\in V} Y_u b(u), \qquad (115)$$

where $a(u, v)$ is the power cost for link (u,v), b(u) is the power cost for switch u, $X_{u,v}$ is a binary decision variable indicating whether link (u,v) is powered ON. $Y_{u,v}$ is a binary decision variable indicating whether switch u is powered ON. The power cost of a link and a switch are considered fixed (there is no such thing as a half-on Ethernet link). The same power model has been used in a work conducted on data center network power consumption optimization by Widjaja *et al.* [232].

Tucker *et al.* [233] described a power model for a global network where they modeled the network as a minimal array of switching devices that connected using a configurable non-blocking multi-stage Clos architecture [234]. Fig. 22 shows the network model they used in their study. In [233] Tucker *et al.* modeled the energy per bit (E_{net}) in the network as,

$$E_{net} = \sum_{i=1}^{s} E_{switch,i} + \sum_{i=1}^{s-1} E_{core,i} + E_{access}, \qquad (116)$$

where E_{net} is the energy consumed by each bit in each stage of switching and in each transport system. $E_{switch,i}$ is the energy per bit in stage i of switching. E_{core} is the energy per bit in a transport system at the output of a switch in stage i of the network in Fig. 22. $E_{transport}$ is the two-way transport energy per bit in the access network. s is the number of stages of switching.

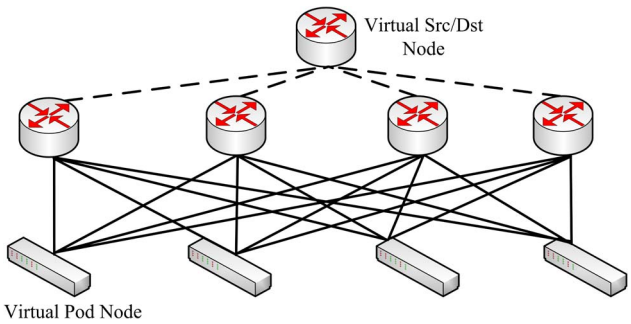


Fig. 23. A core-level subgraph of a 4-ary fat-tree data center network [235]. The diagram includes four pods each of which contains edge switches, aggregation switches, and servers.

In the same line of research of the power model shown in Equation (116), Zhang *et al.* [235] modeled the total power consumption of network switches and links (P^c_{total}) in a core-level subgraph (such as shown in Fig. 23) as,

$$P^c_{total} = \sum_{i=1}^{N^c}\sum_{j=1}^{N^c} x_{i,j}P^L_{i,j} + \sum_{i=1}^{N^c} y_i P^N_i, \qquad (117)$$

where $x_{i,j} \in 0, 1$ and $y_i \in 0, 1$ represent the power status of link $(i, j) \in E$ and node $i \in V$ respectively. $x_{i,j} \in 0, 1$ is a binary variable which is equal to 1 if the link between i and j is powered on or 0 otherwise. Similarly y_i is set to 1 if the node i is powered on or 0 otherwise. Somewhat similar power model for this model has been used for power consumption optimization of a data center network by Jin *et al.* in [236].

Another work which is almost similar to Zhang *et al.* was conducted by Li *et al.* described two power models for data center network traffic flows [237]. Specifically they modeled the total amount of network energy used for transmitting a group of traffic flows in a data center as,

$$E = \sum_{i\in I}\left(P_i t_i + \sum_{j\in J(i)} Q_{i,j}t'_{i,j}\right), \qquad (118)$$

where I and $J(i)$ represent the set of switches and the set of ports in switch i, respectively. The fixed power consumption in switch i is denoted by P_i, while the active working duration of switch i is denoted by t_i. The active working duration of port j in switch i is denoted by $t'_{i,j}$. This general power model was further extended by Li *et al.* by assuming that all the switches run with fixed power P and all the ports have equal power Q. They further assumed the bandwidth capacity of each port as C. With these assumptions, they modeled the total data center network energy consumption as,

$$E = P\sum_{i\in I}\frac{m_i}{U_i C|J(i)|} + Q\sum_{i\in I}\sum_{j\in J(i)}\frac{m'_{i,j}}{CU'_{i,j}}, \tag{119}$$

where m_i and $m'_{i,j}$ represent the aggregate traffic amount traveling through switch i and its port j respectively, while U_i and $U'_{i,j}$ represent the average utilization ratio of switch i and its port j in the transmission duration, respectively.

While there are high level power models being developed to the entire data center networks, there are multiple works conducted focusing on individual network devices deployed in a data center network which is explored next.

2) Modeling Network Device Power Consumption: Data center network is the skeletal structure upon which processors, memory, and I/O devices are dynamically shared and is generally regarded as a critical design element in the system architecture. In this skeletal structure, network devices such as routers, switches, etc. play a pivotal role in data center operations. In most of the situations networking devices such as routers are provisioned for peak loads, yet they operate at low average utilization levels. Router may consume between 80–90% of its peak power when it is not forwarding any packets [238] (i.e., the networking devices display almost no energy proportionality [239]). For example, one of the latest releases of CISCO data center routers Cisco Nexus X9536PQ: 36-port 40 Gigabit Ethernet QSFP + line card consumes 360 W as typical operational power while its maximum power of operation is 400 W [240]. Therefore, measures need to be taken to model the energy consumption of networking devices such as routers which enables development of energy efficient operation techniques. Multiple work have been conducted to model the energy consumption of routers and switches. In this subsection we investigate on the network devices power models based on their level of complexity from least complex to most complex power models.

Additive (componentwise breakdown) power models represent one of the least complicated types of power models for network devices. The simplest power model which can be created for a network device is dividing its power consumption as static and dynamic portions. Energy consumption of a network device operating with a traffic load ρ can be expressed as [241],

$$E(\rho) = E_{static} + E_{dynamic}(\rho), \tag{120}$$

where E_{static} is the static power consumption independent from traffic and $E_{dynamic}(\rho)$ is the dynamic part that is a function of the traffic load ρ.

Another way of constructing additive power models, Vishwanath *et al.* [242] presented the power consumption P of

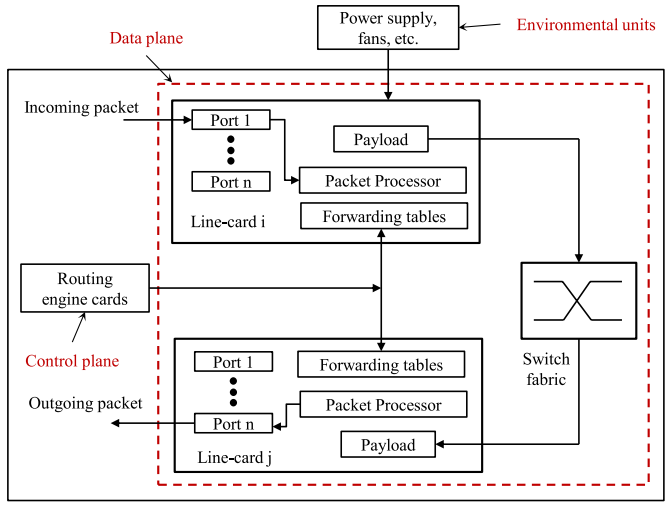


Fig. 24. Structure of a router/switch with control plane, data plane, and environmental units highlighted [242]. Packet processor conducts lookups using the forwarding tables.

an IP router/Ethernet switch as the sum of the power consumed by its three major subsystems (which can be observed clearly in Fig. 24),

$$P = P_{ctrl} + P_{env} + P_{data}, \tag{121}$$

where the terms P_{ctrl}, P_{env}, and P_{data} represents the power consumption of the control plane, environmental units, and the data plane respectively. They further represented the P_{ctrl}, P_{env}, and part of P_{data} which are fixed as P_{idle}. The load dependent component of P_{data} was expanded to two more terms based on packet processing energy and store & forward energy as,

$$P = P_{idle} + E_p R_{pkt} + E_{sf} R_{byte}, \tag{122}$$

where E_p is the per-packet processing energy, and E_{sf} is the per-byte store and forward energy which are constants for a given router/switch configuration. R_{pkt} is the input packet rate and R_{byte} is the input byte rate ($R_{pkt} = \lceil R_{byte}/L \rceil$, where L is the packet length in bytes).

Total energy used by a switch can be modeled in an additive power model as [234], [243],

$$E = \sum_{i=1}^{j} E_{Ii} + E_{supply} + E_{control} - \sum_{j=1}^{k} E_{Oi}, \tag{123}$$

where all the input/output energies to/from the switch are denoted as E_{Ii} and E_{Oi}. The supply and control energies for the switch are denoted as E_{supply} and $E_{control}$ respectively.

A slightly extended version of the energy consumption of network elements can be made by taking the integration of power consumed by the device [244]. The incremental energy (E_{inc}) due to the introduction of an additional traffic flow can be stated as,

$$\begin{aligned} E_{inc} &= \int_{t_1}^{t_2} P\left(C + \Delta C(t)\right) - P(C)dt = \int_{t_1}^{t_2} \Delta P(t)dt, \\ &= \frac{\partial P(C)}{\partial C}\int_{t_1}^{t_2} \Delta C(t)dt = \frac{\partial P(C)}{\partial C} N_{bit} = E_b(C) N_{bit}, \end{aligned} \tag{124}$$

where N_{bit} is the number of transmitted bits and $E_b(C)$ is the energy-per-bit for the network element with throughput C. The incremental power consumption increase was taken to be negligible. The authors mentioned that to use this power model in reality, they need to derive the form of $E_b(C)$ for the given element/elements.

Vishwanath *et al.* presented a methodology for constructing quantitative power models based on vendor neutral measurements on commercially available routers and switches [242]. The system architecture of the router/switch they used in their study is highlighted in Fig. 24.

A linear regression based energy consumption model for residential and professional switches was introduced by Hlavacs *et al.* [245] where the energy consumption $E(P)$ of the device is given by,

$$E(P) \approx \hat{P}(B) = \begin{cases} \alpha + \beta log B, & B > 1, \\ \alpha, & B \leq 1, \end{cases} \tag{125}$$

where α and β correspond to intercept and regression coefficient respectively. B is the bandwidth measured in kbit/s. They conducted measurements with real switches on multiple different aspects and reported the results. For example, they calculated the TCP regression model for a classical 8 port Fast Ethernet switch (Netgear FS608v2) as,

$$\hat{P}_{tcp}(B) = 2.588 - 0.0128 log B. \tag{126}$$

In the same line of research on power consumption modeling of network switches, Mahadevan *et al.* [246] modeled the energy consumption of a network switch as,

$$P_{switch} = P_{chassis} + \alpha P_{linecard} + \sum_{i=0}^{configs} \beta_i P_{configs_i} S_i, \tag{127}$$

where $P_{linecard}$ is the power consumed by linecard with all ports disabled and α is the number of active cards in the switch. The variable *configs* is the number of configurations for the port line rate. $P_{configs\_i}$ is the power for a port operating at speed i while β_i is the number of ports of that category. Here i can be 0, 10 Mbps, 100 Mbps, 1 Gbps, etc. S_i is the scaling factor to account for a port's utilization.

In the line of router power modeling, AHN *et al.* measured the power consumption of an edge router (CISCO 7609) [247]. They found that the power consumption is in direct proportion to the link utilization as well as the packet sizes. Based on this observation they defined basic model for power consumption of a router interface as $P_{interface}(\rho, s, c)$ where ρ means the link utilization, s is the packet size, and c is the computing coefficient which concerns the power consumption overhead by routing protocols. Since the power consumption during the routing protocol exchange is negligible, they simplified their model as $P_{int}(\rho, s)$ which can be stated as,

$$\begin{aligned} P_{int}(\rho, s) &= P_{hp} + P_{pt} \\ &= E_{hp} \times \alpha + E_{pt} \times \beta, \end{aligned} \tag{128}$$

where the header processing power consumption denoted by P_{hp}, packet transferring power consumption denoted by P_{pt}, packet header processing energy is denoted as E_{hp} in Joules, and the per bit transfer energy as E_{pt}(Joule/bit). The data rates of packets per second is denoted by α while bits per second is denoted by β.

3) Power Consumption of Network Interfaces: Network interface card (NIC) is a significant contributor to the system power consumption. Most network hardware operate constantly at maximum capacity, irrespective of the traffic load, even though its average usage lies far below the maximum [248]. Traditional Ethernet is power-unaware standard which uses a constant amount of power independently from the actual traffic flowing through the wires. However, recent high speed Gigabit Ethernet interface cards may consume up to 20 W which makes it reasonable to introduce power saving mechanisms for such network interfaces. Furthermore, in an experiment conducted on TCP energy consumption, Bolla *et al.* observed that their System Under Test (SUT) which was a Linux workstation equipped with 4-core Intel i5 processor, the NIC power consumption varied between (10% to 7%) when the system transitioned from idling to active mode of operation [249]. In both idle and active modes they measured a constant 7 W of power consumed by the NIC. This results in considerable power consumption when running a large scale server installation. Therefore, NIC should be taken into consideration when modeling the overall data center system power consumption.

Network interface card can be either in idle mode or in active mode at any given time [250]. If E_{idle} is the power of the idle interface and $P_{dynamic}$ is the power when active (either receiving or transmitting packets) the total energy consumption of the interface (E_{nic}) can be represented as,

$$E_{nic} = P_{idle} T_{idle} + P_{dynamic} T_{dynamic}, \tag{129}$$

where T_{idle} is the total idle time. $T_{dynamic}$ represents the total active time in a total observation period T. The value of T can be denoted as,

$$T = T_{dynamic} + T_{idle}, \tag{130}$$

where the average NIC power P_{nic} during the period T can be denoted as,

$$\begin{aligned} P_{nic} &= \frac{(T - T_{dynamic}) P_{idle} + P_{dynamic} T_{dynamic}}{T} \\ &= P_{idle} + (P_{dynamic} - P_{idle}) \rho, \end{aligned} \tag{131}$$

where $\rho = T_{dynamic}/T$ is the channel utilization (i.e., normalized link's load). The time periods and the power values depend on the particular network technology employed [250]. The NIC power models described above (in Equations (129) and (131)) are dividing the NIC power consumption as static and dynamic portions.

The choice of network technology could affect utilization of other computer system components (especially CPU) [250]. E.g., In serial point-to-point communications, the CPU is normally used to execute a significant number of communication-related operations which easily increases the dynamic power consumption of CPU. On the other hand embedded network technologies such as Infiniband can move much of the communication work to the embedded architecture. Such behavior can be accommodated in the CPU power models. CPU utilization

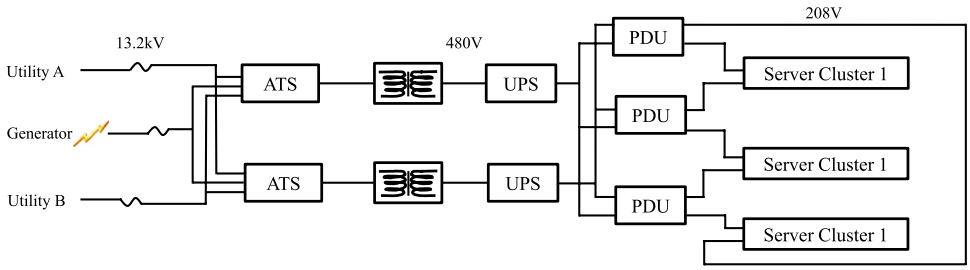


Fig. 25. An example power delivery system for a data center [255]. The system is an example for a high-availability power system with redundant distribution paths.

u, can be denoted as $u' + \gamma\rho$ where u' and $\gamma\rho$ correspond to non-network and network dependent CPU load respectively. $\gamma(\gamma \geq 0)$ models the impact of a given network technology on CPU load based on network utilization ρ. The smaller γ values represent network communications that are dependent on the CPU while larger γ values could be used to model the higher CPU dependency of other network interfaces.

4) Power Consumption of Optical Networks: Optical interconnects provide a viable solution offering high throughput, reduced latency and reduced energy consumption compared to current networks based on commodity switches [251]. In this subsection we describe the optical network power models from the least complicated to most complicated.

Kachris *et al.* [251] created a power model for a reference network architecture designed based on commodity switches. Their reference power model is represented as,

$$P_{ref} = \sum_{racks}(P_{tor} + P_{trans}) + P_{aggrsw} + P_{10gbps}, \tag{132}$$

where P_{tor} is the power consumed by the Top-of-Rack Switch, P_{trans} is the power for the 1 Gbps Ethernet transceivers, P_{aggrsw} is the power of the aggregate switch, and P_{10gbps} is the power of the 10 Gbps Ethernet transceivers. In their model, the energy dissipated by Wavelength Division Multiplexing Passive Optical Network (WDM PON [253]) network for inter track communication was represented as,

$$P_{wdm} = \sum_{racks}(P_{tor} + P_{trans} + P_{sfp}) + P_{aggrsw} + P_{wa}, \tag{133}$$

where P_{sfp} is the power of the optical WDM MMF (multimode fiber [254]) transceivers, P_{wa} is the power of the WDM array port in the aggregate switch. Kachris *et al.* also described power models for six different optical data center network architectures [228]. They described the power consumption of an Arrayed Waveguide Granting Routing (AWGR) based with buffers scheme as,

$$\begin{aligned} P &= \sum P_{trx} + \sum P_{twc} + \sum P_{buffer}, \\ &= nP_{trx} + nP_{twc} + an(P_{oe} + P_{eo} + P_{sdram}), \end{aligned} \tag{134}$$

where P_{trx} is the power of optical transceiver, P_{twc} is the power of the tunable wavelength converter, $P_{shbuffer}$ is the power of the Shared Buffer, $P_{oe,eo}$ is the power of the O/E and E/O converters. P·sdram, n, and a denotes power usage of SDRAM, number of the Top of the Rack switches, and probability of contention respectively. This is an additive power model. A similar technique has been used for modeling the power consumption of the rest of the architectures of which the details can be found at [228].

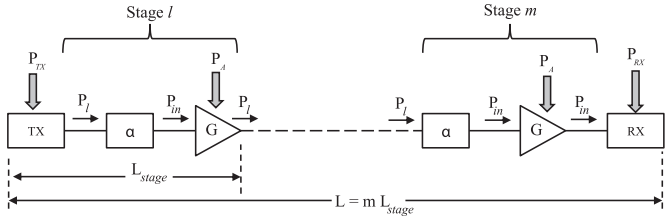


Fig. 26. A WDM transmission system of length L, which comprises an optical transmitter, m identical stages of optical gain, and an optical receiver [233], [243].

The power consumption of a WDM transmission system comprising m identical optically amplified stages as shown in Fig. 26 can be modeled as [243],

$$P_{tot} = mP_A + P_{TX/RX}, \tag{135}$$

where P_A is the supply power to each amplifier and $P_{TX/RX}$ is the supply power to each WDM transmitter/receiver pair. P_{TX} and P_{RX} are the transmitter and receiver supply powers.

Van Heddeghem *et al.* modeled the total power dissipation of an optical multilayer core network (P_{core}) as the sum of its constituting layers,

$$P_{core} = P_{ip} + P_{ethernet} + P_{otn} + P_{wdm}, \tag{136}$$

where the terms P_{ip}, $P_{ethernet}$, P_{otn}, and P_{wdm} represents the power consumption by the IP layer, Ethernet layer, optical transport network, wavelength division multiplexing respectively [252]. Each of these parameters were expanded further as follows,

$$P_{ip} = \delta\left[2\sigma_{ip}\gamma\right], \tag{137a}$$
$$P_{ethernet} = \delta\left[2\sigma_{eth}\gamma\right], \tag{137b}$$
$$P_{otn} = \delta\left[2\sigma_{otn}\gamma\right], \tag{137c}$$
$$P_{optsw} = \delta\left[2\sigma_{oxc}H\right], \tag{137d}$$
$$P_{transponders} = \delta\left[2\sigma_{tr}H\right], \tag{137e}$$
$$P_{amplifiers} = \delta\left[\frac{1}{f}\sigma_{ola}\left\lfloor\frac{\alpha}{L_{amp}}\right\rfloor H\right], \tag{137f}$$
$$P_{regereneration} = \delta\left[\sigma_{re}\left\lfloor\frac{\alpha}{L_{regen}}\right\rfloor H\right], \tag{137g}$$

TABLE VIII
SUMMARY OF POWER CONSUMPTION MODELING APPROACHES FOR DATA CENTER NETWORKS

Work(s)	Type(s)	Characteristics	Limitations
[227][232]	Entire network	Total network energy consumption is expressed as summation of energy dissipated by each network link.	Relatively simple power model.
[237]	Entire network	Represented the total amount of network energy used for transmitting a group of traffic flows in a data center.	Depends on multiple assumptions such as all the switches, ports have fixed power dissipation.
[242]	Network device	Additive power model	Measuring the power consumption by each different units is a challenge.
[244]	Network device	Mathematical integration based power model.	Need additional transformation to use this power model in reality.
[245]	Switch	based on linear regression.	Need to calculate the two constants α and β.
[219]	Switch	An additive power model.	Need to calculate the two constants α and β.
[247]	Router	Based on link utilization and packet size.	The values E_{pt} and E_{hp} have to be calculated in advance.
[250]	Network Interface	Based on static and dynamic power consumptions of network interfaces.	The time values T_{idle} and $T_{dynamic}$ need to be calculated accurately.
[228][251]	Optical Networks	Additive power model.	Depends on the accurate measurement of multiple parameters.
[252]	Optical Networks	Additive power model (total power is calculated as the sum of its constituting layers).	Depends on the accurate measurement of multiple parameters.

where $\gamma = \left(\frac{1}{\eta_{pr}} + H\right)$ of which η_{pr} accounts for traffic protection which equals to 2 for $1+1$ protection. For unprotected traffic the value remains as 1. H represents the average hop count, η_c accounts for the cooling and facilities overhead power consumption in a data center measured by PUE. The value of term δ is given by $\delta = \eta_c \eta_{pr} N_d \overline{D_C}$ where N_d stands for the total number of IP/Multi Protocol Label Switching(MPLS) demands and $\overline{D_C}$ is the average demand capacity.

A summary of power consumption modeling approaches for data center networks is shown in Table VIII. Next, we move onto describing one of the most important, yet non-IT component of a data center: the power conditioning unit.

C. Modeling Energy Consumption of Power Conditioning Systems

Power conditioning system of a data center is responsible of delivering electric power to the loads of the system (IT and mechanical equipment). Maintaining adequate power quality levels and consistency of power supply is a must [256]. Power conditioning system of a data center consumes significant amount of energy as the power wasted during transformation process which can be traced in its power hierarchy. Distribution of uninterrupted electrical power into a data center requires considerable infrastructure (such as transformers, switchgears, PDUs, UPSs, etc. [257]). In a typical data center power hierarchy, a primary switch board distributes power among multiple Uninterrupted Power Supply sub-stations (UPSs). Each UPS in turn, supplies power to a collection of PDUs. A PDU is associated with a collection of server racks and each rack has several chassis that host the individual servers. Such an arrangement forms a power supply hierarchy within a data center [258], [259]. An illustration of such power hierarchy is shown in Fig. 25 [255], [260], [261].

PDUs are responsible for providing consistent power supply for the servers. They transform the high voltage power distributed throughout the data center to voltage levels appropriate for servers. PDUs incur a constant power loss which is proportional to the square of the load which can be represented as [262], [263],

$$P_{pdu\_loss} = P_{pdu\_idle} + \pi_{pdu}\left(\sum_{N} P_{srv}\right)^2, \tag{138}$$

where P_{pdu_loss} represents power consumed by the PDU, while π_{pdu} represents the PDU power loss coefficient, and P_{pdu_idle} which is the PDU's idle power consumption. The number of servers in the data center is represented by N.

UPSs on the other hand act as the temporary power utilities during power failures [262]. Note that in different data center designs UPS can sit before PDU or it can sit in between PDU and the server(s). UPSs incur some power overheads even when operating on utility power which can be modeled as,

$$P_{ups\_loss} = P_{ups\_idle} + \pi_{ups}\left(\sum_{M} P_{pdu}\right), \tag{139}$$

where π_{ups} denotes the UPS loss coefficient. Pelley *et al.* mentioned that PDUs typically waste about 3% of their input power while for UPSs it amounts for 9% of UPS input power at full load. Next, we describe the power modeling efforts related to data center cooling systems.

D. Modeling Data Center Cooling Power Consumption

Even a carefully designed, racked blade system using low-voltage components can consume up to 22 kW of power [264]. These levels of power consumption generate considerable heat that has to be disposed in order for servers to operate within a safe operating temperature range. Cooling systems are used to effectively maintain the temperature of a data center [265]. Cooling power is the biggest consumer of the non-computing power in a data center followed by power conversion and other losses [266]–[268]. Fig. 27 provides a breakdown of the cooling power in a data center [82]. The data center cooling power is a function of many factors such as layout of the data center,

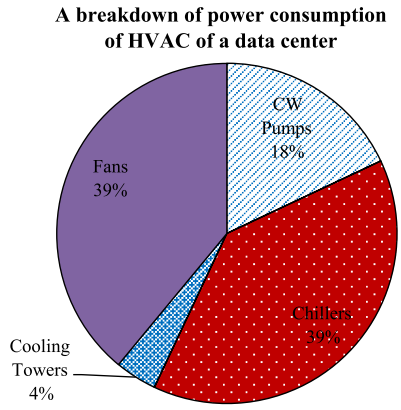


Fig. 27. An example power breakdown of a data center HVAC [82]. The three major power consumers in the HVAC includes fans (39%), chillers (39%), and cooling water pumps (18%).

the air flow rate, the spatial allocation of the computing power, the air flow rate, and the efficiency of the CRAC units. In this subsection first we investigate on the server (i.e., system) level fan power models. Next, we describe the CRAC unit level power models which can be considered as an extension of the fan power models. Finally, we list down some of the power modeling work that cannot be specifically categorized into the aforementioned two categories.

1) Power Consumption of System Fans: Cooling fans located inside a system unit represents another important energy consumer of a data center server. For example, one of the studies made by Vasan *et al.* have shown that the fans consumed about 220.8 W (18%) while the CPU power consumption was measured at 380 W (32%) [269], while the total approximate system power consumption was measured at 1203 W. In another study conducted by Lefurgy *et al.* it was observed that fan power dominated the small configuration of IBM p670 server power envelope (51%) while in large configuration it represented a considerable portion (28%) of the server power envelope [270]. Furthermore, fan power is a cubic function of fan speed ($P_{fan} \propto s_{fan}^3$) [16], [271]. Hence over-provisioning of cold air into the servers can easily lead to energy inefficiencies [272]. These statistics indicate that power consumption by system fans is of considerable amount that need to be accounted in modeling the power hierarchy of a modern data center.

In one of the most simplest fan power models cooling power can be expressed as a function of the IT power as [217],

$$f_a(d) = kd^3, \tag{140}$$

where $0 \leq d \leq \overline{d}$, $k > 0$. The parameter k depends on the temperature difference $(t_{ra} - t_{oa})$ which is based on the heat transfer theory. Here t_{oa} is the outside air temperature and t_{ra} is the temperature of the (hot) exhausting air from the IT racks. $\overline{d}$ is the maximum capacity of the cooling system that can be modeled as $\overline{d} = C(t_{ra} - t_{oa})$.

In an extended version of the power model shown in Equation (142), Meisner *et al.* [273] described the Computer Room Air Handler (CRAH) power consumption as,

$$P_{crah} = P_{idle} + P_{dyn} f^3, \tag{141}$$

where f is the fan speed of CRAH (between 0 to 1.0) and P_{idle} and P_{dyn} represent the idle and dynamic power usage of CRAH unit.

In certain literature, fan power consumption is added to an aggregated quantity called "Electromechanical Energy Consumption" [92]. It is typical that multiple fans exist in a server. The power drawn by the ith fan at time t can be denoted by,

$$P_{fan}^i(t) = P_{base} \left(\frac{RPM_{fan}^i(t)}{RPM_{base}} \right)^3, \tag{142}$$

where P_{base} is the base power consumption of the unloaded system without running any applications (Note that the RPM_{base} corresponds to the fan's rpm while running the base workload). This value is obtained in [92] by measuring the current drawn on the +12V and +5V lines using a current probe and an oscilloscope. Therefore, if there are total N fans installed in the server, the total electromechanical energy consumption (E_{em}) over a given task execution period of T_p is denoted by,

$$E_{em} = \int_0^{T_p} \left(\sum_{i=1}^{N} P_{fan}^i(t) \right) dt, \tag{143}$$

where the $P_{fan}^i(t)$ is denoted by the power model in Equation (142).

Unlike other components of a computer system, the system fans have received less attention from power modeling researchers. Mämmelä *et al.* measured the power consumption of system fans by directly attaching cabling to the measurement device [274]. Based on the performance results they obtained, they constructed a power model for system fans as,

$$\begin{aligned} P_{fan} = {} & 8.3306810^{-15} a^4 + 8.51757\omega^4 - 2.9569d^4 \\ & - 1.1013810^{-10} a^3 + 54.6855\omega^3 - 76.4897d^3 \\ & + 4.8542910^{-7} a^2 + 258.847\omega^2 - 1059.02d^2 \\ & - 6.0612710^{-5} a + 32.6862\omega + 67.3012d - 5.478, \end{aligned} \tag{144}$$

where ω denotes the fan width (in mm), d denotes the fan depth (in mm), and a presents the revolutions per minutes. This is a fourth order, relatively complicated polynomial power model.

2) CRAC Power Models: Typically, the largest consumer of power and the most inefficient system in a data center is CRAC. Factors that affect such operation of the CRAC unit include the operational efficiency and the air distribution design of the unit [275]. Percentage of the cooling power varies, but can be up to 50% or more in a poorly designed and operated data center [276], [277]. About 40% of the total energy consumption in the telecom industry is devoted to cooling equipment in data centers [278]. Most of this energy is consumed by the site chiller plant, CRAC, and by the air handlers (CRAH [279]) [280]. Heat dissipation in a data center is related to its server utilization [281]. Studies have shown that for every 1W of power utilized during the operation of servers an additional 0.5–1 W of power is consumed by the cooling equipment to extract the heat out from the data center [118]. Power consumption of the data

center cooling equipment generally depends on two parameters, first the amount of heat generated by the equipment within the data center and second due to the environmental parameters such as temperature [282]. Here we organize the CRAC power modeling efforts from least complicated to most complicated power models.

One of the simplest power models for CRAC was described by Zhan *et al.* They partitioned the total power budget among the cooling and computing units in a self consistent way [283]. They modeled the power consumption c_k of a CRAC unit k as,

$$c_k = \frac{\sum_i p_i}{\eta}, \tag{145}$$

where $\sum_i p_i$ is the power consumption of servers with their heat flow directed towards the CRAC unit. η is the Coefficient of Performance (CoP). Based on the physical measurements they have created an empirical model for η of a commercial water-chilled CRAC unit as,

$$\eta = 0.0068t^2 + 0.0008t + 0.458, \tag{146}$$

where t is the supply air temperature of the CRAC unit in degrees Celsius.

Moore *et al.* [220] modeled the cooling power consumption as,

$$C = \frac{Q}{\eta(T = T_{sup} + T_{adj})} + P_{fan}, \tag{147}$$

where Q is the amount of server power consumption, $\eta(T = T_{sup} + T_{adj})$ is the η at $T_{sup} + T_{adj}$. Note that T_{sup} is the temperature of the cold air supplied by the CRAC units. They assumed a uniform T_{sup} from each CRAC unit. T_{adj} is the adjusted CRAC supply temperature. η is the coefficient of performance which gives the performance of the CRAC units. η is the ratio of heat removed (Q) to the amount of work necessary (W) to remove that heat which can be expressed as $\eta = Q/W$ [220]. A higher η value shows a more efficient process which requires less work to remove a constant amount of heat. P_{fan} is the total power consumed by the CRAC fans. A similar power model for modeling the cooling power consumption is described in [284].

Additive power models can be observed in data center cooling power models as well. In one such work Das *et al.* developed models for power consumption of cooling support systems. Their model included the computer room air conditioning fan, refrigeration by chiller units, pumps of the cooling distribution unit, lights, humidity control and other miscellaneous items [285]. They modeled the total power dissipation of a data center (P_{rf}) [286] as,

$$P_{rf} = P_{it} + P_{pdu} + P_{crac} + P_{cdu} + P_{misc}, \tag{148}$$

where P_{rf} corresponds to the raised floor power, P_{it} is the power consumed by the IT equipment. P_{crac} is the power consumed by computer room air conditioning units. The power losses happen due to uninterruptible power supply (UPS) systems and losses associated with the power distribution are represented as P_{pdu}. They used P_{cdu} to denote the power dissipation for the pumps in the cooling distribution unit (CDU) which provide direct cooling water for rear door and side-car heat exchange mounted on a few racks. This model is almost equal to the model of raised floor power described by Hamann *et al.* in [287] where the latter has a term P_{light} to denote the power used for lighting. The total CRAC power consumption and total CDU power can be denoted as follows,

$$P_{crac} = \sum_i P_{crac_i}, \text{ and } P_{cdu} = \sum_j P_{cdu_j}, \tag{149}$$

where i and j correspond to CRAC and CDU respectively. The CRAC system used in their study equipped with variable frequency drivers (VFDs) which showed the following empirical relationship between fan power P_{crac_i} and relative fan speed θ_i for a respective CRAC,

$$P_{crac_i} = P_{crac_i,100}(\theta_i)^{2.75}, \tag{150}$$

where $P_{crac_i,100}$ is the fan power at $\theta_i = 100\%$. Furthermore, they showed that under steady state conditions (i.e., after thermal equilibrium is reached) the energy balance requires that the total raised floor power (P_{rf}) equal the total cooling power (P_{cool}), that is provided by both the $crac_s$, $P_{cool(crac)}$, and the rear-door/side-car heat exchanger or CDU, $P_{cool(cdu)}$. Therefore, raised floor power P_{rf} can be denoted as,

$$P_{rf} = P_{cool} = \sum_i P_{cool(crac_i)} + \sum_j P_{cool(cdu_j)}. \tag{151}$$

The cooling power of CRACs and CDUs can be denoted as the product of the fluid flow rate in cfm (Cubic feet per minute), the temperature differential (δT_i) between the cold fluid emerging from the unit and the hot fluid returned back to the unit, and the density and specific heat of the fluid. Therefore, these two quantities can be denoted as,

$$P_{cool(crac_i)} = \phi_{crac_i}\Delta T_i/3293\,[cfm^{\circ}\text{F/kW}], \tag{152}$$

$$P_{cool(cdu_j)} = \phi_{cdu_j}\Delta T_j/6.817\,[cfm^{\circ}\text{F/kW}]. \tag{153}$$

Furthermore, they showed that since all raised floor power needs to be cooled by the chilling system, which requires power for refrigeration (P_r) that can be approximated as,

$$P_{chiller} = P_r/\eta, \tag{154}$$

where η is the coefficient of performance of the chiller system described earlier of this section. They assumed a value of 4.5 for η which they mentioned to be somewhat typical for large-scale centrifugal chilling systems based on the case study results of a large scale data center described in [288].

Certain power models such as the one described by Kaushik *et al.* for a data center in [289] express the power consumption of a system as a ratio. Cooling power consumption increase (from P_1 to P_2) due to the requirement of air-flow increase within the data center from V_1 to V_2 can be represented as follows [276], [290],

$$\frac{V_2}{V_1} = \frac{R_2}{R_1} = \left(\frac{P_2}{P_1}\right)^3, \tag{155}$$

where R_1 and R_2 correspond to the rounds per minute (RPM) values of the fan.

TABLE IX
SUMMARY OF DATA CENTER COOLING POWER DISTRIBUTION MODELING APPROACHES

Work(s)	Characteristics	Limitations
[217]	Cooling power is expressed as a function of IT power.	Need to calculate the parameter k which is based on the heat transfer theory.
[273]	Additive power model of CRAH unit.	Depends on multiple assumptions.
[92]	Fan power consumption is expressed in terms of RPM values.	Depends on multiple assumptions.
[274]	System fan power model based on curve fitting.	Relatively complicated polynomial.
[283]	Total power budget of a system was partitioned among cooling and computing units in a self consistent manner.	Depends on multiple assumptions such as the value of η which is the coefficient of performance.
[220]	Focused on the fan power and is based on the coefficient of performance (CoP)	Assumes uniform supply temperature from each CRAC unit.
[285][286]	Introduced a series of power models.	Depends on multiple assumptions.
[289]	Considers the air flow within a data center.	Measurement of accurate air flow within a data center is challenging.
[291]	Considers time factor.	Depends on multiple assumptions.
[276][289][290]	Power consumption of a system is expressed as a ratio.	Depends on multiple assumptions.
[292]	Considered the scenario where cooling unit can be turned-off	Depends on multiple assumptions.

Certain power models take into account the temporal features of data center power usage. In one such work Tu *et al.* described a data center total power consumption [291] as the sum of the server, power conditioning system, and the cooling system power draw, that can be expressed as a time-dependent function of $b(t)$ $(b(t) = f_s(x(t), a(t)))$,

$$b(t) + f_p\left(b(t)\right) + f_c^t\left(b(t)\right) \triangleq g_t\left(x(t), a(t)\right), \tag{156}$$

where $x(t)$ is the number of active servers and $s(t) \in [0, x(t)]$ is the total server service capability at time t. To get the workload served in the same time slot $s(t) > a(t)$. They also described a similar power model for water chiller cooling system as,

$$f_c^t\left(b(t)\right) = Q_t b^2(t) + L_t b(t) + C_t, \tag{157}$$

where $Q_t, L_t, C_t \geq 0$ depend on outside air and chilled water temperature at time t.

In a similar power model, Zheng *et al.* described a power consumption model CRAC systems [292]. They summarized the total power consumption of cooling system as,

$$P_{cooling_j} = \begin{cases} \alpha_j U_j^2 + \beta_j U_j + \gamma_j + \theta_j, & \text{if } U_j \leq 25\%, \\ \theta_j, & \text{Otherwise,} \end{cases}, \tag{158}$$

where θ is the power consumption made by CRAC (this was simplified as a fixed power consumption). α_j, β_j, and γ_j correspond to the chiller power consumption coefficients in the data center j. U_j is the system utilization (%) in data center j. If the total workload in the data center j (U_j) is less than 25% of the total data center processing capacity, all chillers can be turned off to save cooling system energy. This is the reason why such division of the cooling energy happens in their power model.

Similar to the use of PUE in the previous section, it can be observed that multiple power models have been developed concentrating the *CoP* metric. A summary of data center cooling power distribution modeling techniques is shown in Table IX.

E. Metrics for Data Center Efficiency

High levels of energy has been consumed by data centers to power the IT equipment contained within them as well as to extract the heat produced by such equipment. Data center industry's heavy reliance on power has historically triggered the requirement for use of metrics for tracking the operational efficiency of data centers [293], [294]. In this section we describe few key power consumption metrics used for measuring the energy efficiency of data centers. Such description is needed because certain data center power consumption models have used these metrics. In this section we organize the metrics based on their significance for energy efficiency measurement from most significant to least significant.

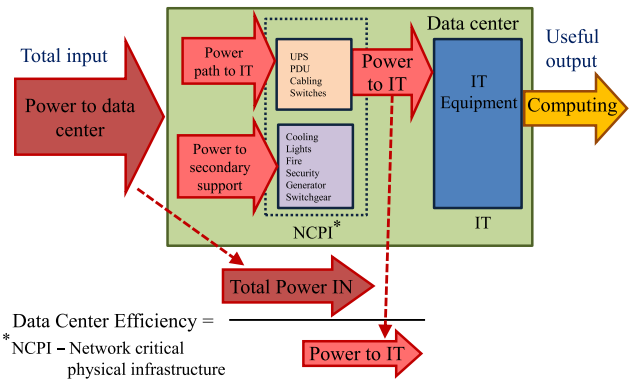


Fig. 28. An illustration of how PUE is defined [263]. Data center efficiency is defined as the fraction of input power delivered to the IT load.

One of the most widely used data center energy efficiency metric is *Power Usage Effectiveness (PUE)* [295], [296]. It is a metric to compare different data center designs in terms of their electricity consumption [297] (See the illustration in Fig. 28 [263]). PUE of a data center (η_{pue}) is calculated as,

$$\eta_{pue} = \frac{\text{Total data center annual energy}}{\text{Total IT annual energy}}, \tag{159}$$

where the *Total data center annual energy* is the sum of power drawn by cooling, lightening, and IT equipment. PUE is a value greater than 1 ($\eta_{pue} \geq 1$) since data centers draw considerable amount of power as non-IT power. Google data centers reported a PUE of 1.12 in 2013 [298]. A higher PUE translates into a greater portion of the electricity coming to the data center spent on cooling and the rest of the infrastructure (A visual explanation is available in [299]). While PUE is widely

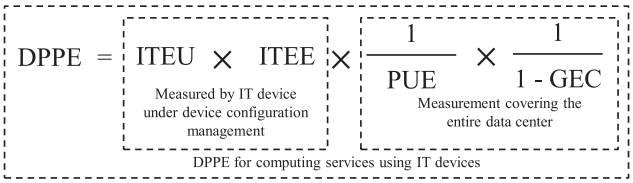


Fig. 29. Relationship between DPPE, ITEU, ITEE, PUE, and GEC metrics [309]. DPPE is designed to increase as the values of these four indicators increase (the inverse value for PUE).

recognized as the preferred energy efficiency metric for data centers [300], a good PUE value is not enough to guarantee the global efficiency of the data center, because PUE metric does not consider the actual utilization (applications and workloads [301]) of computational resources [302], [303]. Furthermore, typical PUE reports communicate the minimum data center infrastructure power use. Hence, can only be used to determine the minimum potential energy usage of the corresponding facility [304].

Another metric used in data center energy efficiency measurement is *Data Center Infrastructure Efficiency (DCiE)* which is expressed as follows [41], [305],

$$\eta_{dcie} = \frac{1}{\eta_{pue}} = \frac{\text{IT Devices Power Consumption}}{\text{Total Power Consumption}} \times 100\%. \tag{160}$$

DCiE is the reciprocal measurment for PUE. While both these metrics are used by data center vendors, PUE has been used more commonly than DCiE [306].

Patterson *et al.* described two new data center energy consumption metrics called *IT-power usage effectiveness (ITUE)* and *Total-power usage effectiveness (TUE)* [307]. ITUE is defined as total IT energy divided by computational energy. TUE is the total energy into the data center divided by the total energy to the computational components inside the IT equipment. TUE can be expressed as product of ITUE and PUE metrics.

Data center Performance Per Energy (DPPE) is a metric which indicates the energy efficiency of the data center as a whole. DPPE is a metric for indicating data center productivity per unit energy [308]–[310]. DPPE (η_{dppe}) is defined as follows,

$$\eta_{dppe} = \frac{\text{Throughput at the data center}}{\text{Energy consumption}}. \tag{161}$$

Another performance related to the data center energy consumption is the *Green Energy Coefficient (GEC)* [309]. In simple terms GEC is the ratio between the green energy (e.g., wind power, solar power, etc.) consumed by the data center and the total data center energy consumption. In practice, PUE and GEC are measured for the entire data center level where as ITEU and ITEE are measured only for some measurable IT devices. The relationship between the aforementioned data center energy consumption metrics can be shown as in Fig. 29.

Data Center Energy Productivity (DCeP) is a metric introduced by Sego *et al.* for measuring the useful work performed by a data center relative to the energy consumed by the data center in performing the work [311]. Therefore, DCeP (η_{dcep}) can be expressed as,

$$\eta_{dcep} = \frac{W}{E_{total}} = \frac{\text{Useful work produced}}{\text{Total energy consumed by the data center}}, \tag{162}$$

where the *Useful work produced* is measured through physical measurements. The total energy consumed is gathered during an interval of time called the assessment window. DCeP metric allows the user to define the computational tasks, transactions, or jobs that are of interest, and then assign a measure of importance of economic value to each specific unit of work completed.

Similar to higher level data center energy consumption efficiency metrics, several power related metrics have been proposed for measuring data center cooling system efficiency. The *Data Center Cooling System Efficiency (CSE)* characterizes the overall efficiency of the cooling system (which includes chillers, pumps, and cooling towers) in terms of energy input per unit of cooling output [294], [312]. The CSE (η_{cse}) can be expressed as,

$$\eta_{cse} = \frac{\text{Average cooling system power usage}}{\text{Average cooling load}}. \tag{163}$$

Data Center Workload Power Efficiency (DWPE) was proposed as a complete data center energy efficiency metric by Wilde *et al.* [313]. DWPE (or simply d) was proposed as the first metric that is able to show how energy efficient a HPC system is for a particular workload when it is run in a specific data center. DWPE (η_d) is defined as,

$$\eta_d = \frac{\eta_p}{\eta_u}, \tag{164}$$

where η_p is a performance per Watt metric for a HPC system while η_u is the system PUE which is a metric that defines the effectiveness of the system in a specific data center.

F. Modeling Energy Consumption of a Data Center

As described in Section I, a data center is a container that holds multiple items such as groups of servers, storage, networking devices, power distribution units, and cooling systems. This section describes the energy consumption modeling work conducted at the whole data center level which accounts for energy consumed by all the aforementioned components. We list the power models in this section in a two fold manner. First, we investigate on the general power models for data centers. Next, we investigate on a specific category of power models which are developed based on the PUE of the data center.

Perhaps the most simplest types of the energy consumption models that can be created for an entire data center would be such as the one described by Aebischer *et al.* [314]. They mentioned that modeling the electricity demand of a set of devices does not require a complex model, but needs much input data while not all of the available data are statistically significant. They modeled the energy consumption in the use

phase of a collection of devices by following a bottom-up approach,

$$E(t) = \sum_{ijk} n_i(t) \times e_{ij}(t) \times u_{ijk}(t), \tag{165}$$

where n is the number of devices of type i, e is the power load in functional state j and u is the intensity of use by user k. A similar data center power model was created by LeLou: *et al.* [315] where they expressed the power consumption of a data center as the sum of the minimum (i.e., idle) power consumptions of its hosts and the consumptions induced by the VMs. However, both such works do not consider the non-IT power consumption of a data center.

CPU utilization based power models can be observed even at the higher levels of data center hierarchy. In one of this category of works, Raghavendra *et al.* constructed power/performance models for data centers based on CPU utilization [316]. For each system in the data center they calibrated models on the actual hardware by running workloads at different utilization levels and measuring the corresponding power and performance (in terms of the percentage of the work done).

Islam *et al.* described a utilization based power model [317] for an entire data center power consumption as,

$$P\left(u(t), x(t)\right) = \sum_{i=1}^{M} p_i\left(u_i(t), x_i(t)\right), \tag{166}$$

where $x(t) = (x_1(t), x_2(t), \ldots, x_M(t))$ and $u(t) = (u_1(t), u_2(t), \ldots, u_M(t))$ are the vectors of speed selections and utilization of data center servers respectively. In their model i corresponds to a server. They have ignored the toggling costs such as turning a server off, VM migration, etc. Similar work to this research can be observed in [318] and [319].

PUE (described in Section VIII-E) is heavily utilized for modeling the power consumption of data center systems since PUE represents the total IT power consumption of a data center. From here onwards we organize the PUE based data center power models in chronological order. One such power models for entire data center was created by Masanet *et al.* They described an electric power consumption model for data centers as follows [320],

$$E^d = \sum_j \left[\sum_i E^s_{ij} + E^s_j + E^n_j \right] \eta_{pue_j}, \tag{167}$$

where E^d represents the data center electricity demand (kWh/y), E^S_{ij} is the electricity used by servers of class i in space type j (kWh/y), E^s_j is the electricity used by external storage devices in space type j (kWh/y), E^n_j is the electricity used by network devices in space type j, and η_{pue_j} is the power utilization effectiveness of infrastructure equipment in space type j. They mentioned that their model estimates data center energy demand as a function of four variables that account for the electricity use of servers, external storage devices, network devices, and infrastructure equipment. This is another example (similar to the previous power model by Mahmud *et al.* in Equation (169)) for calculation of the total data center power by multiplying total IT power by PUE. However, this power model is an additive power model which differentiates it from the power models described in Equations (169) and (171).

In the same way Yao *et al.* extended their server level power model [121], [321] for modeling the power consumption of an entire data center as,

$$P\left(N_i(t), b_i(t)\right) = \left(N_i(t) \left(\frac{b_i(t)^\alpha}{A} + P_{idle} \right) \right).U, \tag{168}$$

where the A, P_{idle}, and α parameters have the same meaning as in Equation (29). The use of U term accounts for additional power usage due to cooling, power conversion loss, etc. for having $N_i(t)$ servers active.

Mahmud *et al.* [322] mathematically denoted the total power consumption of a data center during time t by $p(\lambda(t), m(t))$, that can be expressed as,

$$p\left(\lambda(t), m(t)\right) = \eta_{pue} \sum_{j=1}^{J} m_j(t) \left[e_0 + e_c \frac{\lambda_j(t)}{m_j(t)\mu_j} \right], \tag{169}$$

where $\eta_{pue} > 1$ is the PUE, e_0 is the static server power irrespective of the workloads. Here e_c is the computing power incurred only when a server is processing workloads. $\lambda_j(t)$ is the arrival rate of type-j jobs. $m_j(t)$ is the number of servers for type-j jobs. The service rate of a server for processing type-j jobs is μ_j. The $\lambda(t) = (\lambda_1(t), \ldots, \lambda_J(t))$, and $m(t) = (m_1(t), \ldots, m_J(t))$. Their power model can be considered as an extension of a power model for a group of servers (described in Section VIII-A) to an entire data center via use of PUE metric.

In another similar line of research Zhou *et al.* described a data center power model using the power usage efficiency metric (PUE) [323]. Their model is a hybrid of the power model for a single server (which is almost similar to the one in Equation (30)) and data center PUE. Given the number of active servers $m_j(t)$, parameters α_j, β_j, ν_j and power usage efficiency metric PUE_j in data center j, the power consumption of data center j in time slot t can be quantified by $E_j(t)$ as,

$$E_j(t) = PUE_j m_j(t) \left[\alpha_j \mu_j^{\nu_j}(t) + \beta_j \right], \tag{170}$$

where α is a positive factor, β is the power consumption in idle state, ν is an empirically determined exponent parameter ($\nu \geq 1$) with a typical value of $\nu = 2$.

In a similar line of research was presented by Liu *et al.* [324] where they created the power model by combining the workload traces and the PUE (η_{pue}) to create the total power demand of a data center as,

$$\nu(t) = \eta_{pue}(t)\left(a(t) + b(t)\right), \tag{171}$$

where $a(t)$ is the power demand from the inflexible workload and $b(t)$ is the power demand from the flexible workload.

IX. Software Energy Models

Up to now we have focused on energy consumption models based on physical characteristics of the data center. But equally important is to consider the type of applications and workloads

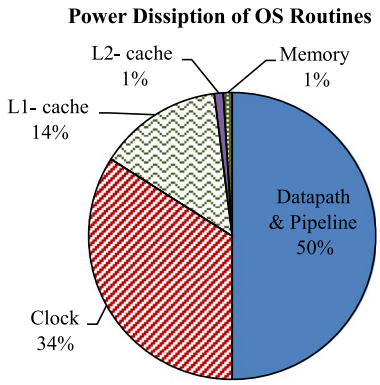


Fig. 30. Power dissipation breakdown across OS routines [146]. Data-path and pipeline structures that support multiple issue and out-of-order execution are found to consume 50% of total power on the examined OS routines.

a data center handles. Data center software can be broadly categorized into five categories: *compute-intensive*, *data-intensive*, *communication-intensive applications*, and *OS and virtualization software*, and general software. In the computation-intensive tasks the major resource consumer is the CPU core(s). In a data-intensive task the storage resources in the cloud system are the main energy consumer. In communication-intensive tasks a large proportion of the energy is used by network resources such as network cards, routers, switches, etc. In the following subsections, we explore the energy consumption modeling efforts in the context of OS and virtualization, data-intensive, communication-intensive, and computation-intensive tasks, as well as general data center applications.

A. Energy Consumption Modeling at the OS and Virtualization Level

An operating system (OS) sits in between the two key layers of the data center stack: the physical hardware and the applications. Much of earlier work on energy efficiency has focused on modeling the energy usage of hardware components or software applications. Applications create the demand for resources and physical hardware are the components that actually consume the IT power. The OS was largely considered an intermediary between the two key layers. This section first lists the power consumption models of OSs and then moves on to describing the power models for VMs. Furthermore, we list the power models in order from simpler to more complex.

It is important to understand which OS level events gives rise to power consumption before starting to model OS power usage. One such characterization was done by Li *et al.* [146], which characterized the behavior of a commercial OS across a large spectrum of applications to identify OS energy profiles. The OS energy consumption profiling gave a breakdown of power dissipation of OS routines as shown in Fig. 30. According to this chart, the data-path and pipeline structures which support multiple issues and out-of-order execution are found to consume 50% of total power on the examined OS routines. The capacitive load to the clock network which switches on every clock tick also causes significant power consumption (about 34% in Fig. 30). The number of instructions that flow through a data-path usually determines its energy consumption. Furthermore the Instruction Level Parallelism (ILP) performance measured by Instructions Per Cycle (IPC) impacts the circuit switching activities in the microprocessor components and can result in significant variations in power. Based on these observations Li *et al.* created the following simple linear regression model for OS routine power consumption as,

$$P = k_0 + k_1\psi. \tag{172}$$

Here, k_0 and k_1 are regression model parameters. The ILP is denoted by ψ. This power model was extended to the entire OS energy consumption model as follows,

$$E_{OS} = \sum_i (P_{osr,i}T_{osr,i}). \tag{173}$$

Here, $P_{osr,i}$ is the power of the i'th OS routine invocation and $T_{osr,i}$ is the execution time of that invocation. They mentioned that $P_{osr,i}$ can be computed in many ways, for example by averaging the power usage of all invocations of that routine usage in a program.

Once the routines with high power consumption are identified, the related high level OS performance counters can be utilized to construct OS level power models. Davis *et al.* presented composable, highly accurate, OS-based (CHAOS) full-system power models for servers and clusters [325]. These power models were based on high-level OS performance counters. They evaluated four different modeling techniques of different conceptual and implementation complexities. In their method the full system power is represented as a function $\hat{f}()$ of high level OS performance counters represented by $(x_1, \ldots, x_n)$. In each model they varied the number of model features, starting from CPU utilization to the full cluster specific and general feature sets. Their baseline linear power model can be shown as,

$$\hat{f}() = a_0 + \sum_i a_i x_i, \tag{174}$$

where the parameters $(a_i)_0^n$ are fitted by minimizing the squared error. This baseline model is used to compare all other proposed power models for $\hat{f}(x_1, \ldots, x_n)$ and to evaluate the increase in accuracy of more complex models. They create the following piecewise linear power model as,

$$\hat{f}() = a_0 + \sum_i \sum_j a_{i,j} B_{i,j}^s(x_i, t_{i,j}). \tag{175}$$

This model provides an extra degree of freedom where the parameter s can be positive $(+)$ or negative $(-)$, and the basis functions $B_{i,j}^s$ are hinge functions such that,

$$B_{i,j}^+(x,t) = \begin{cases} 0, & \text{if } x = t, \\ x - t, & \text{otherwise}, \end{cases} \tag{176}$$

$$B_{i,j}^-(x,t) = \begin{cases} 0, & \text{if } x > t, \\ t - x, & \text{otherwise}, \end{cases} \tag{177}$$

where t thresholds are called *knots* and the j indices permit a feature to be responsible for multiple knots. The authors mentioned that fitting these models requires finding the knots $t_{i,j}$ and the parameters $a_{i,j}$. They used an implementation of the *Multivariate Adaptive Regression Splines* (MARS) algorithm for this purpose. They mentioned that these models can express a feature such as CPU utilization which may consume different amounts of full-system power in different regions of operation. They also proposed a quadratic model which extends the piecewise linear model and introduces nonlinearity within each segment by making the basis functions interact. This quadratic power model [326] can be represented as,

$$\hat{f}() = a_0 + \sum_i \sum_j a_{i,j} B_i^s(x_i, t_i) B_j^s(x_j, t_j), \qquad (178)$$

where the model restricts the interaction among the basis functions to a degree of two. They used the same MARS algorithm to select knots and fit parameters to select which bases would interact. The most complicated power model they introduced was a switching model which can be given as,

$$\hat{f}() = I(f)\left(a_0 + \sum_i a_i x_i\right) + (1 - I(f))\left(a_0' + \sum_i a_i' x_i'\right), \qquad (179)$$

where $I(f) = 1 \iff x_i <$ threshold; otherwise $I(f) = 0$. This switching power model uses CPU frequency in an indicator function $I(f)$, allowing each p-state/frequency to have its own linear model. This results in a set of (possibly) different linear models depending on the clock frequency. The switching model's indicator function partitions the space for all the features, creating completely separate models for each frequency state. They also mentioned that the switching model is more rigid even though it may require more parameters and may have discontinuities at the knots (i.e., frequency transitions) [325].

Application checkpoint-restart is an important technique used by operating systems to save the state of a running application to secondary storage so that it can later resume its execution from the state at which it was checkpointed [327]. Power models built for OS processes (especially in the context of data centers) need to consider the energy consumed by such checkpoint-restart mechanisms. Coordinated checkpointing periodically pauses tasks and writes a checkpoint to stable storage. The checkpoint is read into memory and used to restart execution if a CPU attached to a socket fails. The power models for an operating system process created by Mills *et al.* defined the total energy consumption of a single process which uses checkpoint and restart as,

$$E_{cpr} = E_{soc}(\sigma_{max}, [0, T_\omega]) + E_{io}([0, \delta]) \times \frac{T_s}{\tau} + E_{io}([0, R]) \times \frac{T_\omega}{M_{sys}}, \qquad (180)$$

where the first term $E_{soc}(\sigma_{max}, [0, T_\omega])$ correspond to the energy consumed by a socket (i.e., CPU) at speed σ_{max} (the maximum execution speed of the CPU), during a check-point restart period of length T_ω. The model assumes at any given time all processes are either working, writing a checkpoint or restoring from a checkpoint and all sockets are always executing at σ_{max}. The second portion of the equation adds the energy required to write or restore from a checkpoint times the number of times the process will be writing or recovering from a checkpoint. M_{sys}, τ, T_s, δ, and R stand for the system's MTBF, checkpoint interval, check point time, and recovery time respectively.

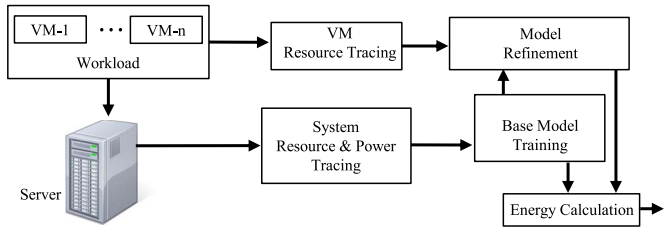


Fig. 31. Joulemeter VM power metering [88]. Joulemeter is intended to provide the same power metering functionality for VMs similar to hardware meters for physical servers.

In the next half of this subsection we discuss works to model the power usage of virtual machines. We list the power models in the order of increasing complexity. The models have several types, e.g., power models associated with software frameworks, temporal power models, component based (additive) power models, and models based on the state of operation of a VM such as VM live migration.

Software frameworks have been developed to estimate the power consumption of VMs. The work by Kansal *et al.* developed power models to infer power consumption from resource usage at runtime and identified the challenges which arise when using such models for VM power metering [88]. Once the power models are developed they can be used for metering VM power by tracking each hardware resource used by a VM and converting the resource usage to power usage based on a power model for the resource. They mentioned that their approach does not assume the availability of detailed power models from each hardware component as required by some previous works on VM energy usage measurement [328]. They proposed a mechanism for VM power metering named Joulemeter (See Fig. 31 for details).

Some of the notable components of Joulemeter include the resource tracing module which uses the hypervisor counters to track the individual VM resource usage. Through experiments Kansal *et al.* demonstrated that linearity assumptions made in linear power models do lead to errors. They also stated that the magnitude of the errors was small compared to full system energy but was much larger compared to the energy used by an individual VM. They also mentioned that errors reported were averaged across multiple different workloads and can be higher on specific workloads. Kansal *et al.* mitigated such errors by using built-in server power sensors that were not available in the older servers used in prior works.

Temporal aspects such as the number of events occurring in a particular time window can be used to create VM power models. In one such example work on VM power modeling, Kim *et al.* created a power model for VMs assuming the power consumption of the VMs in time period t is determined by the

number of events that occur during t [329]. Energy consumption of a VM in their model is represented as,

$$E_{vm_{i,t}} \propto C_1 N_i + C_2 M_i - C_3 S_t, \tag{181}$$

where the coefficients C_1, C_2, and C_3 are obtained by conducting multi-variable linear regression over data sets sampled under diverse circumstances. N_i is the number of retired instructions during the time interval t, M_i is the number of memory accesses, and S_t is the number of active cores at time t.

Componentwise power consumption breakdowns can be obtained in the context of VMs as well. One way to do this is to break down power consumption as either static or dynamic [330]. Then the total power consumption of a VM can be expressed as,

$$P_{vm_i}^{total} = P_{vm_i}^{idle} + P_{vm_i}^{dynamic}, \tag{182}$$

where $P_{vm_i}^{idle}$ is the static/idle power consumption while $P_{vm_i}^{dynamic}$ is the dynamic power consumption. In [330] the main focus was on modeling the idle power consumption of a VM. The power model is expressed as,

$$P_{vm_i}^{idle} = \begin{cases} P_{server_j}^{idle}, & \Leftrightarrow \exists k / U_{vm_i}^{res_k} = 100\%, \\ \frac{\sum_k \alpha_k U_{vm_i}^{res_k}}{\sum_k \alpha_k} P_{server_j}^{idle}, & \text{otherwise}, \end{cases} \tag{183}$$

where the weight assigned to the resource k (possibly 1) is denoted by α_k and the utilization of resource k by vm_i is represented as $U_{vm_i}^{res_k}$. The model expresses the fact that the idle power consumption of a given VM is equal to the idle power consumption of the server on which it runs if and only if the VM uses 100% of any of the server/hypervisor's resources (e.g., disk, RAM, or the number of the virtual CPUs (vCPUs)) because this prevents the server from hosting another VM. In other situations, the VM's idle power consumption is correlated to its utilization of each resource. Weights are used to account for the resource scarcity since resources such as RAM, vCPU, etc. might limit the number of hosted VMs in the server.

Another approach for VM power modeling is to break down power usage into components such as CPU, memory, IO, etc.

$$P_{vm} = \alpha U_{cpu} + \beta U_{mem} + \gamma U_{io} + e, \tag{184}$$

where U_{cpu}, U_{mem}, and U_{io} represent CPU utilization, memory usage, and disk IO throughput, respectively. e is an adjustment value. The weights α, β, and γ need to be trained offline. Note that this is almost similar to the power model described in Equation (10), which also represented server power using a componentwise breakdown.

VM live migration is a technology which has attracted considerable interest from data center researchers in recent years [22]. VM live migration is a very important tool for system management in various scenarios such as VM load balancing, fault tolerance, power management, etc. VM migration involves source host, network switch, and destination hosts. Liu *et al.* presented an energy consumption model for VM migration as follows [22],

$$E_{mig} = E_{sour} + E_{dest} = (\alpha_s + \alpha_d) V_{mig} + (\beta_s + \beta_d), \tag{185}$$

where $\alpha_s, \alpha_d, \beta_s$, and β_d are model parameters to be trained. V_{mig} is measured in megabytes and the energy E_{mig} is measured in joules. The authors mentioned that the model can be used with heterogeneous physical hosts as well. In this case the model parameters need to be retrained for each of two different platform. In a homogeneous environment the modeling equation reduces to $E_{mig} = \alpha V_{mig} + \beta$. They learned the energy model parameters by using linear regression, and the model was given as $E_{mig} = 0.512 V_{mig} + 20.165$.

B. Modeling Energy Consumption of Data-Intensive Applications

Data-intensive tasks are usually I/O bound and they require processing large volumes of data [331]. Data-intensive applications can be categorized as *online data-intensive* [332] and *offline data-intensive* applications based on their type of operation. It can be observed that most of the current data-intensive application power models can be categorized under the second type, a significant percentage of which are MapReduce power models. Therefore, in this subsection, we first delve into the details of power consumption modeling of general data-intensive applications, before considering power models of MapReduce applications that are heavily deployed in current data centers.

A data warehouse is an example of an offline data-intensive application that gets frequently deployed in data center clusters. Poess *et al.* developed a power consumption model for enterprise data warehouses based on the TCP-H benchmark [333]. The simplified power consumption model they developed can be applied to any published TPC-H result and is representative data warehouse systems. However, this model is intended to estimate peak power consumption of the system [333]. They described the power consumption of the entire server as,

$$P_s = (C_c P_c + 9 C_m + C_{di} P_d) * 1.3 + 100, \tag{186}$$

where P_s is the power consumption of the entire server, C_c is the number of CPUs per server, P_c is the Thermal Design Power (TDP) of a CPU in watts ($P_c \in [55, 165]$ in the processors used in their study), C_m is the number of memory DIMMs per server, C_{di} is the number of internal disks per server, and P_d is the disk power consumption. They also added 30% of the power usage of the above components plus 100 watts to the above model to account for the power overhead of the chassis. Furthermore, they described the power consumption of the I/O subsystem (P_{io}) as,

$$P_{io} = C_e * C_{de} * P_{de} * 1.2, \tag{187}$$

where C_e is the number of enclosures, C_{de} is the external disks per enclosure, P_{de} is the the power consumption of the external disk ($P_{de} \in [7.2, 19]$ in the external disks used in their study). They added 20% of the power as the overhead of the enclosure. Then they expressed the power consumption of the entire system as $P = P_s + P_{io}$.

Many data-intensive applications such as text processing, scientific data analysis, and machine learning can be described as a set of tasks with dependencies between them [334]. These

applications are called *workflow* applications and are designed to run on distributed computers and storage. Energy consumption of a data-intensive workflow execution was described by Gamell *et al.* as [335],

$$E = P_{node,idle}Nt + E_{computation} + E_{motion}, \quad (188)$$

where t is the execution time. The term $P_{node,idle}Nt$ represents the total energy consumption by the idling nodes. They modeled the energy consumption during the computation phase of the workflow as,

$$E_{computation} = \frac{P_{cpu,dynamic}}{C} I_s V \left(t_{prod,v} + \frac{t_{cons,v}}{I_a} \right), \quad (189)$$

where V, I_s, and I_a represent number of variables, number of simulation steps, and number of simulation steps between two analyses respectively. The two time related parameters $t_{prod,v}$ and $t_{cons,v}$ represent the time taken to produce a variable (s) and the time taken to consume a variable s, respectively. Since the workflow involves the use of a deep memory hierarchy, the power model needs to consider the power consumption during the data loading and storage phases. Similarly energy consumption for data motion was defined as,

$$E_{datamotion} = VI_s \left(\sum_{\beta \in mem,stg,net} \left(t^{st}_{v,\beta} + t^{ld}_{v,\beta} \right) P_{\beta,dyn} \right), \quad (190)$$

where the model is constructed by multiplying VI_s with the summation of three sub terms corresponding to staging (stg), network access (net), and memory access (mem). The two terms $t^{ld}_{v,\beta}$ and $t^{st}_{v,\beta}$ represent data loading and storage times, respectively.

MapReduce is one of the most frequently used data processing models in today's data centers. Furthermore, there has been multiple recent works on the energy efficiency of MapReduce and Hadoop applications [336], [337]. We now discuss some of this work. MapReduce is a programming model for processing and generating large data sets [338]. With the widespread adoption of the MapReduce programming model through implementations such as Hadoop [339] and Dryad [340], MapReduce systems has become one of the key contributors to modern data center workloads.

Regression techniques have been utilized to create MapReduce power models. One example for such work is done by Zhu *et al.* [341], [342] where they developed a general power consumption model for each node i in a Hadoop cluster as follows,

$$p_i(k) = A_i p'_i(k-1) + B_i \Delta x_i(k), \quad (191)$$

where A_i and B_i are known system parameters which may vary due to the varying workloads. p'_i is the measured power consumption [343]. The change in the arrival rate threshold for node i is given by Δx_i. The k'th control point represents the time k. [342] used a recursive least square (RLS) estimator with exponential forgetting to identify the system parameters A_i and B_i for all nodes i. They used this power model in a power aware scheduler which they implemented for Hadoop (see Fig. 32).

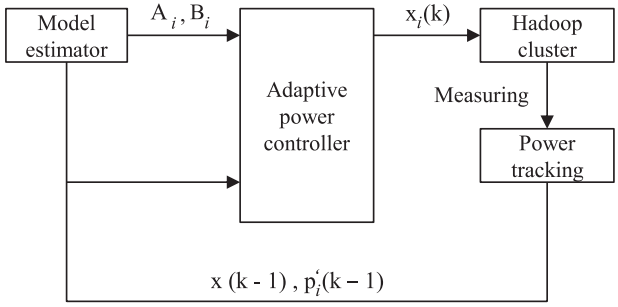


Fig. 32. Workflow of the admission controller. The model estimator component dynamically models the power consumption of each server [343].

The model estimator dynamically models the power consumption of each server to ensure the accuracy under the dynamic workloads. For managing the power peaks, the controller module makes control decisions based on the model generated by the model estimator. This is an example for an application of the energy consumption modeling and prediction process described in Section I of this paper.

Additive approaches have been used in MapReduce power models as well. In one such work, Feng *et al.* [344] presented an energy model for MapReduce workloads as,

$$E = PT = P_i T_i + P_m T_m + P_s T_s + P_r T_r, \quad (192)$$

where energy consumed by the MapReduce cluster is represented by multiplying the power P with time T. This is modeled in more detail by summing the energy consumed for job initialization ($P_i T_i$), the map stage ($P_m T_m$), reduce stage ($P_r T_r$), and the intermediate data shuffling ($P_s T_s$). Furthermore, they mentioned that there are four factors that affect the total energy consumption of a MapReduce job, namely the CPU intensiveness of the workload, I/O intensiveness, replica factor, and the block size.

Another additive power model for a MapReduce cluster was presented by Lang *et al.* [345]. In their model the total energy consumption $E(\omega, \upsilon, \eta)$ is denoted as,

$$E(\omega, \upsilon, \eta) = (P_{tr} T_{tr}) + \left(P^{n}_{\omega} + P^{\bar{n}}_{\omega} \right) T_{\omega} + \left(P^{m}_{idle} + P^{\bar{m}}_{idle} \right) T_{idle}, \quad (193)$$

where P_{tr} is the average transitioning power, where transitioning refers to turning nodes on and off. Transitioning power can have a significant impact on the energy consumption of a MapReduce cluster. T_{tr} is the total transitioning time in υ, $P^{[n,\bar{n}]}_{\omega}$ is the on/off-line workload power, $P^{[n,\bar{n}]}_{idle}$ is the on/off-line idle power. Variables n and m correspond to the number of online nodes running the job, and the number of online nodes in the idle period. Variables $\bar{n} = N - n$ and $\bar{m} = N - m$ are the corresponding offline values. Furthermore, the time components for $E(\omega, \upsilon, \eta)$ must sum to υ, where,

$$\upsilon = T_{tr} + T_{\omega} + T_{idle}. \quad (194)$$

Since the workload characteristics may require the job be run within some time limit τ, the cluster energy management problem can be cast as,

$$min\ (E(\omega, \upsilon, \eta))\ | T_{\omega} \leq \tau. \quad (195)$$

Through the Equation (193) it can be observed that energy consumption reduction can be achieved by powering down parts of the cluster. Furthermore, it was shown that reducing the power drawn by online idle nodes P^{m}_{idle} can have a big impact on energy management schemes.

C. Modeling Energy Consumption of Communication-Intensive Applications

Communication-intensive applications are composed of a group of tasks during the course of a computation exchange a large amount of messages among themselves [346]. Communication-intensive applications are generally network bound and impose a significant burden on the data center network infrastructure. Most such applications are developed using the Message Passing Interface (MPI), a widely used API for high performance computing applications [347]. This subsection lists some of the notable efforts in communication-intensive data center application power modeling.

Message broadcasting is a typical communication pattern in which data belonging to a single process is sent to all the processors by the communicator [347]. Diouri *et al.* presented techniques for energy consumption estimation of MPI broadcast algorithms in large scale HPC systems. Their methods can be used to estimate the power consumption of a particular broadcast algorithm for a large variety of execution configurations [348]. In the estimator component of their proposed technique they used two models of energy consumption for modeling the MPI Scatter And Gather (MPI/SAG) algorithms and the Hybrid Scatter And Gather (Hybrid/SAG). Since both models follow similar structure, we show the MPI/SAG model below,

$$\begin{aligned} E_{sag} &= \sum_{i=1}^{N} \xi_{sag}^{node_i} + \sum_{j=1}^{M} \xi_{sag}^{switch_j} \\ &= t_{scatter}(p,N)\left(\sum_{i=1}^{N} \rho_{scatter}^{node_i}(p) + \sum_{j=1}^{M} \rho_{scatter}^{switch_j}\right) \\ &\quad + t_{allgather}(p,N)\left(\sum_{i=1}^{N} \rho_{allgather}^{node_i}(p) + \sum_{j=1}^{M} \rho_{allgather}^{switch_j}\right), \end{aligned} \tag{196}$$

where the model represents the total energy consumption as a sum of energy consumption by the nodes and switches. Furthermore, the work expanded the first line of the Equation (196) by incorporating the time factor and splitting the energy consumption of the scatter and gather phases into two separate subterms, which are symmetric to each other. It should be noted that the variable p in the energy model corresponds to the number of processes per node. N is the number of compute nodes, and M is the number of switches.

In their work on exploring data-related energy/performance tradeoffs at extreme scales, Gamell *et al.* used machine-independent code characteristics (e.g., data access and exchange patterns, computational profiles, messaging profiles, etc.) to develop a power model, which was then validated empirically using an instrumented platform [349]. They followed a similar approach for dividing the energy consumption among different components of a computer system as described in Equation (11). They approximated the dynamic processor memory and power dissipation using an activity factor/switching activity (α),

$$P_{system}^{dynamic} = \alpha_{cpu} P_{cpu}^{active} + \alpha_{mem} P_{mem}^{active}, \tag{197}$$

where P_{cpu}^{active} corresponds to the dynamic power consumption of a CPU, as described in Equation (7). The activity factors are computed from the number of operations per second (*MIPS*), number of memory accesses per second (mem_{bw}), and a normalization factor that represents the maximum capacity as follows,

$$\alpha_{cpu} = \frac{mips}{\max mips}; \alpha_{mem} = \frac{mem_{bw}}{\max mem_{bw}}. \tag{198}$$

Unlike some of the previous works, they modeled the energy consumption that occurs during communication (E_{comm}) between MPI processes as follows,

$$E_{comm} = \begin{cases} \sum_{i=1}^{M} \frac{data_i}{BW_{net}} P_{transfer}, \\ \qquad \text{if } smp(src_i) \neq smp(dest_i), \\ \sum_{i=1}^{M} \frac{data_i}{BW_{mem}} \left(P_{cpu}^{active} + P_{mem}^{active}\right), \\ \qquad \text{if } smp(src_i) = smp(dest_i), \end{cases} \tag{199}$$

where $smp(i) = smp(j)$ is used to indicate that the MPI ranks i and j are mapped to cores that share memory. BW_{net} and BW_{net} are the bandwidth values of the network and memory respectively. $P_{transfer}$ depends on the network characteristics, e.g., whether the network is InfiniBand or Gemini.

D. Modeling Energy Consumption of General Applications

There has been a number of works on data center application power modeling which cannot be specifically attributed to one of the four types of applications described above. This section lists such power models, starting from more general power modeling approaches such as performance counter based software power models, algorithmic power models, and the application of software architecture concepts [350] for power consumption modeling. Next, we describe more specific power models such as web service power models and business processes, which are frequently deployed in data centers.

Multiple software based techniques and APIs have been developed in recent years for inferring individual component level power consumption. Some of these software APIs utilize external power meters for measuring the system power consumption and conducts online analysis of the measured data. For example, PowerPack is a collection of software components that enables correlating system component level power profiling to application functions [56], [70]. PowerPack depends on power meters connected to the hardware system to obtain measurements. PowerPack allows the user to obtain direct measurements of the major system components' power,

TABLE X
SUMMARY OF SOFTWARE ENERGY MODELS

Work(s)	Category	Characteristics	Limitations
[333]	OS	Based on linear regression.	Regression model parameters k_0 and k_1 need to be calculated beforehand.
[325][326]	OS	Based on linear regression but more complicated than [333].	Regression model parameters need to be calculated beforehand.
[327]	OS	Considers the power consumed by processes during the process of application checkpointing.	Based on multiple assumptions.
[329]	VM	Temporal aspects such as the number of events occurring at a particular time window is considered.	The constants C_1, C_2, and C_3 need to be obtained by multi-variable linear regression.
[330]	VM	Componentwise power consumption breakdown	Power values of the components need to be known beforehand.
[22]	VM	Considers VM live migration scenario.	Multiple model parameters need to be trained.
[333]	Enterprise data warehouses	Based on TCP-H benchmark.	Intended for estimating the peak power. Does not consider data warehouse specific aspects.
[60]	distributed applications	Based on Application Profiles	Based on multiple assumptions. $\beta_1 ... \beta_4$ need to be calculated in advance.
[355][356]	Software Architectural Styles	Derived cost models for five different architectural styles	The framework is fairly high level one.
[348]	MPI	Estimates the power consumption of a particular broadcasting algorithm.	Depends on multiple assumptions such as portability of the algorithm.
[349]	extreme scale applications	Based on machine-independent code characteristics.	Depends on multiple assumptions. Value of α needs to be calculated in advance.
[335]	workflow	Additive power model.	Depends on multiple assumptions.
[358]	General software applications	Based on application throughput.	Depends on multiple assumptions. α and β constants need to be calculated beforehand.
[351]	General software applications	Additive power model.	Need to calculate $a_1 ... a_5$ in advance. Depends on the accuracy multiple parameters such as cache miss rate, context switching rate, etc.
[357]	General software applications	The total energy consumption of a compute nodes of a job.	Depends on multiple assumptions.
[359]	Algorithms	A processor based power model.	Depends on multiple assumptions.
[341][342]	MapReduce	Regression based power model.	Multiple parameters such as A_i, B_i, and δx_i need to be calculated beforehand.
[344][345][352]	MapReduce	Additive power models. [345] does not consider MapReduce specific aspects.	Depends on multiple assumptions.
[360][361]	Web services	A detailed power model.	Depends on multiple assumptions.
[362]	Business processes	An additive power mode which calculates the power consumed by each process instance.	This power model is for process instance i of the entire business processes.

including CPU, memory, hard disk, and motherboard power usage.

Another category of this type of software infers system power consumption using the system specification. Instruction level energy profiles can be utilized to create energy profiles for specific target platforms. In one of such power modeling work, Smith *et al.* described "Application Profiles," which is a means for presenting resource utilization of distributed application deployment. The Application Profiles they developed captures usage of the CPU, memory, hard-disk, and network accesses. They described "CloudMonitor," a tool that infers the power consumption from software alone through the use of computationally generated power models [60]. They mentioned that since servers in a data center are normally procured in batches with the same configuration, training of the model is only required on one server per batch. The resulting model is able to predict power usage across the remaining servers without the need for using dedicated power meters. They mentioned that the power model is applicable to different workloads if the hardware configuration is the same across multiple machines. Their power model can be expressed as,

$$P = \alpha + \beta_1 P_{cpu} + \beta_2 P_{mem} + \beta_3 P_{hdd} + \beta_4 P_{net}, \quad (200)$$

where the model considers each hardware subcomponent that they measured and their approach generates weights automatically during the training phase. Here α is the baseline power and the coefficients β_1, β_2, β_3, and β_4 represent the coefficients for the power consumption of the CPU (P_{cpu}), memory (P_{mem}), HDD (P_{hdd}), and network (P_{net}), respectively.

A similar additive power model for the energy consumption of data center applications was described by Aroca *et al.* [351]. Another work on modeling software power consumption by Wang *et al.* created an additive power model as [352],

$$P = a_1 cpu_u + a_2 \gamma + a_3 \delta + a_4 \sigma + a_5 cpu_f + P_{idle}, \quad (201)$$

where $a_1, \ldots, a_5$ are a set of coefficients to be determined by a set of training benchmarks. The parameters cpu_u, γ, δ, σ, cpu_f, and P_{idle} represent the CPU utilization, cache miss rate, context switching rate, instructions per cycle, CPU frequency, and idle power dissipation of the system, respectively.

Similarly *CoolEmAll* project [353] (which is aimed at decreasing energy consumption of data centers by allowing designers, planners, and administrators to model and analyze energy efficiency of various configurations and solutions) takes in to account multiple factors when creating data center application power models. They used application level estimator based

on performance counters, model specific registers (MSR), and system information for this purpose [354]. In total they used 17 different variables of the aforementioned three categories.

While individual application power models can be constructed as described above, higher level abstractions for software systems can be created by considering their software architecture. In one such example, Seo *et al.* described a framework that supports early estimation of the energy consumption induced by an architectural style in a distributed system. Their framework defines a method to derive platform and application-independent equations that characterize a style's energy consumption behavior. They derived energy cost models for five architectural styles: peer-to-peer, C2, client-server, publish-subscribe (pub-sub), and pipe-and-filter [355]. They also described a framework for estimating energy consumption of Java-based software systems [356]. Their energy cost model consists of linear equations. Their energy consumption model for a distributed system is an additive model where the total system energy consumption (E_{total}) is denoted by its constituent n components and m connectors as,

$$E_{total} = \sum_{i=1}^{n} E_i + \sum_{j=1}^{m} C_j, \tag{202}$$

where E_i denotes the energy consumption of the component i and C_j denotes the energy consumption of the connector j. This power model was further expanded to a generic energy consumption model, which we do not describe in this paper. Interested readers can refer to [355] for more details.

Another power model which did not consider the application specific details was described by Cumming *et al.* [357]. They denoted the total energy consumed by the compute nodes of a job as,

$$E = \frac{E_n + N/4 \times 100W \times \tau}{0.95}, \tag{203}$$

where $N/4 \times 100 \times \tau$ accounts for 100 W-per-blade contribution from a network interconnect. τ is the wall clock time for running the application. The denominator 0.95 is used to adjust for AC/DC conversion.

In a different line of research, Koller *et al.* investigated about an application-aware power model [358]. They observed that the marginal (dynamic) power for any application A_i has a linear relationship with application throughput (λ_i). They proposed an application throughput based power model as,

$$P(A_i) = \alpha_i + \beta_i \lambda_i, \tag{204}$$

where α and β are constants for each application which need to be measured in separate calibration runs for each application and on each server type the application is placed on. These two parameters can be inferred using two calibration runs. This power model does not have any correspondence to the general power model described in this paper. Although power model is abstract, Koller *et al.* state that the actual slope for more than 90% of operational systems had less than 5% error, indicating that throughput based power modeling is quite accurate.

Similar to the algorithmic power consumption models described in Equations (54), (87), and (210), Demmel *et al.* described a technique for modeling the total energy cost E of executing an algorithm [359]. In their model they sum the energy costs of computation (proportional to the number of flops F), communication (proportional to the number of words W and messages S sent), memory (proportional to the memory used M times the run time T) and "leakage" (proportional to runtime T) for each processor and multiplied by the number of processors p. This power model can be expressed as,

$$E = p(\gamma_e F + \beta_e W + \alpha_e S + \delta_e MT + \epsilon_e T), \tag{205}$$

where δ_e is the energy cost per stored word per second. γ_e, β_e and α_e are the energy costs (in joules) per flop, per word transferred and per message, respectively. The term $\delta_e MT$ assumes that energy is used only for memory that are used for the duration of the algorithm (which is a strong architectural assumption). ϵ_e is the energy leakage per second in the system outside the memory. ϵ_e may encompass the static leakage energy from circuits as well as the energy of other devices not defined within the model (e.g., disk behavior or fan activity).

Web services power modeling is one of the more specific types of application power modeling scenarios. Bartalos *et al.* developed linear regression based models for the energy consumption of a computer system considering multiple aspects such as number of instructions executed, number of sent or received packets, CPU cycles (CPU unhalted events), IPC, percentage of non-idle CPU time, and last level cache misses [360], [361]. They estimate the computer's instantaneous power consumption while executing web service workloads using an aggregate linear instantaneous power model.

Business processes in the form of web services are frequently deployed in data center systems. Nowak *et al.* modeled the power consumption of such business processes [362], defining the power consumption of a complete process instance as,

$$P_i = \sum_{j=1}^{m} (C_i(j)) + E, \tag{206}$$

where $C_i(j)$ is the power consumption of an activity a. The power consumed by the process engine performing the activity is given by E and $j = (1, \ldots, m)$ is the number of activities of a business process model. Note that this power model is only for a single process instance i of an entire business process consisting of I total business processes. A summary of software of energy consumption models is shown in Table X.

X. Energy Consumption Modeling Using Machine Learning

Machine learning (ML) is a scientific discipline which is concerned with developing learning capabilities in computer systems [27]. In this section we provide a brief introduction to machine learning. Next, we describe the use of machine learning techniques in the context of data center power modeling by describing on use of supervised, unsupervised, reinforcement,

TABLE XI
SUMMARY OF MACHINE LEARNING BASED ENERGY CONSUMPTION PREDICTION APPROACHES

Work	Category	Algorithm(s)	Characteristics	Limitations
[363][367]	Supervised	Decision Trees (M5P)	The model predicts multiple energy related parameters.	Networking costs have not been addressed.
[101]	Unsupervised	GMM	Average error of less than 10% across different workloads.	I/O related workload metrics are not considered.
[373]	RL	Q-Learning	Based on model-free constrained reinforcement learning	Depends on the number of heuristics for training the model.
[378]	RL	B-ANN	An embedded software power model.	Less portable.
[379]	RL	ANN, LR	Used ANN and LR for resource overbooking	Time consuming ANN training process.
[140][374][376]	RL	MLP	Predicts overbooking ratios for cloud computing systems.	Outliers in power consumption results in low quality predictions.

and evolutionary learning algorithms in the context of data center power modeling respectively (Table XI).

A. Machine Learning - An Overview

A computer system is said to learn if it improves its performance or knowledge due to experience and adapts to a changing environment. Results from machine learning are in the form of information or models (i.e., functions) representing what has been learned. The results of what has been learned are most often used for making predictions, similar to the use of manually created models described in the first half of this paper.

In recent years the use of machine learning techniques for power consumption modeling and prediction has been a hot topic among data center energy researchers. Different types of algorithms that are prominent in machine learning and data mining can be applied for prediction of power consumption in a data center [363]. Machine learning algorithm should be computationally lightweight and should be able to produce good results when trained with various workloads.

Machine learning algorithms can generally be categorized under four themes: supervised learning, unsupervised learning, reinforcement learning, and evolutionary learning [364]. In this section of the paper we follow a similar categorization to summarize the energy consumption prediction research conducted using machine learning. However, some power prediction research constitutes the use of multiple different machine learning techniques, and cannot be placed in a specific category.

B. Supervised Learning Techniques

The most common type of learning is supervised learning. In supervised learning algorithms, a training set of examples with correct responses (targets) is provided. Based on the training set, the algorithm generalizes to respond correctly to all possible inputs. Algorithms and techniques such as linear regression, nonlinear regression, tree based techniques (such as classification trees, regression trees, etc.), support vector machines (SVM), etc. are all supervised learning techniques. Most of the work on linear regression and non-linear regression has been discussed in the previous sections. In this section we discuss some of the other supervised learning techniques.

A decision tree is a supervised learning technique using a tree of decision nodes [365]. Decision trees break classification down into a set of choices about each feature, starting from the root of the tree progressing down to the leaves, where the classification decision is given. The M5 algorithm is the most commonly used classifier in this family. Berral *et al.* presented a methodology for using machine learning techniques to model the main resources of a web-service based data center from low-level information. They used the M5P algorithm [366] for calculating the expected CPU and I/O usage [363], [367]. M5P is the implementation of M5 algorithm in the Weka toolkit [368]. It uses a decision tree that performs linear regressions on its leaves. This is effective because CPU and I/O usage may differ significantly in different workloads, but are reasonably linear in each. They use normal linear regression to model memory. The work first models virtual machine (VM) and physical machine (PM) behaviors (CPU, memory, and I/O) based on the amount of load received. The input data for the analysis are,

- The estimated requests per time unit.
- The average computational time per request.
- The average number of bytes exchanged per request.

Then high-level information predictors are learned to drive decision-making algorithms for virtualized service schedulers, without much expert knowledge or real-time supervision [369], [370]. The information collected from system behaviors was used by the learning model to predict the power consumption levels, CPU loads, and SLA timings to improve scheduling decisions [371]. The M5P algorithm was used because simple linear regression is incapable of describing the relationship between resources and response time.

They learned the following function which can predict the estimated effective resources required by a VM based only on its received load without imposing stress on the VM or occupation on the PM or network,

$$f(l) \rightarrow E[\mu_{cpu}, \mu_{mem}, \mu_{io}], \tag{207}$$

where l, μ_{cpu}, μ_{mem}, and μ_{io} represent the load, CPU utilization, memory utilization, and amount of I/O, performed respectively. They also learned a function that calculates the expected response time from placing a VM in a PM with a given occupation such that the scheduler can consolidate VMs without excessively increasing the response time,

$$f(s, r) \rightarrow E[\tau], \tag{208}$$

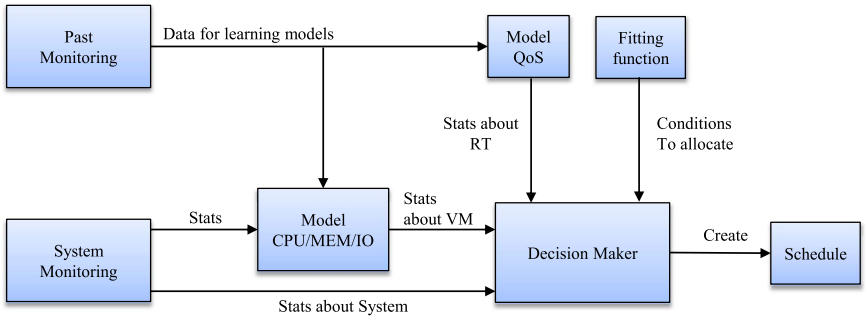


Fig. 33. Flow of decision making with machine learning. The process involves learning models from online system performance information as well as from empirical information.

where τ represents the run time (RT), s represents the status, and r represents the resources. The overall decision making system is shown in Fig. 33.

Rules based learning is another type of supervised learning, and are a popular alternative to decision trees [366] which are also categorized under supervised learning [365]. Decision tress can be easily turned into a set of if-then rules which are suitable for use in a rule induction system [364]. During the conversion process one rule is generated for each leaf in the decision tree. Fargo *et al.* applied reasoning to optimize power consumption and workload performance by mapping the current system behavior to the appropriate application template (AppFlow type [372]), defined as,

$$A_{type} = f(u, v, n), \tag{209}$$

where the three parameters CPU utilization (u), memory utilization (v), and processor number (n) are used to determine the AppFlow type (A_{type}).

C. Unsupervised Learning Techniques

Unlike supervised learning, a training set of examples with correct responses are not provided in unsupervised learning [364]. Clustering algorithms (such as hierarchical clustering, k-means clustering) and Gaussian mixture models (GMM) are examples for unsupervised learning techniques.

Power models can be created to correlate power consumption to architectural metrics (such as memory access rates and instruction throughput) of the workloads running in the VMs. The metrics collection can be conducted per VM and can be fed to the model to make the power prediction [101]. Such systems are non-intrusive because they do not need to know the internal states of the VMs and the applications running inside them. The model uses a GMM based approach. The GMMs are trained by running a small set of benchmark applications. Dhiman *et al.* implemented this technique on a computer running Xen virtualization technology [101]. They showed that their approach can perform online power prediction with an average error of less than 10% across different workloads and different utilization levels.

D. Reinforcement Learning Techniques

Reinforcement learning (RL) algorithms have a behavior which is in between supervised and unsupervised learning [364]. There is a significant amount of uncertainty and variability associated with the energy consumption model using information coming from the environment, application, and hardware. An online power management technique based on model-free constrained reinforcement learning was presented in [373] as a solution. In this work, the power manager learns a new power control policy dynamically at runtime from the information it receives via RL.

Neural networks have been widely used for predictions about resource overbooking strategies, with the goal of achieving more efficient energy usage. In one of the earliest works applying this technique, Moreno *et al.* implemented a Multilayer Perceptron (MLP) neural network to predict the optimum amount of computing resources required by a customer's applications based on historical data [140]. The MLP neural network based resource predictor processes the customer's utilization data to predict the resource consumption of the current submitted workload.

One of the more recent works on resource overbooking is *iOverbook*, an autonomous, online, and intelligent overbooking strategy for heterogeneous and virtualized environments [374]. A similar neural network based technique for power modeling was used by Guzek *et al.* [375].

When using neural networks based energy consumption prediction it is important to evaluate multiple different neural networks with different characteristics, e.g., different numbers of inputs. Tesauro *et al.* followed a similar approach for developing control policies for real-time management of power consumption in application servers [376]. They developed a 2-input and 15-input neural network model a state-action value function defining the power manager's control policy in an IBM BladeCenter cluster. They observed that the 15-input neural network with preprocessing exhibits the steadiest response time, while the power cap [377] decisions of 15-input neural network showed quite large short-term fluctuations.

Li *et al.* analyzed the relationship between software power consumption and some software features on the algorithmic level [378]. They measured time complexity, space complexity, and input scale, and proposed an embedded software power model based on algorithm complexity. They designed and trained a back propagation artificial neural network (B-ANN) to fit the power model accurately using a sample training function set and more than 400 software power data points.

There have been works that integrate reinforcement learning with other machine learning techniques for power consumption

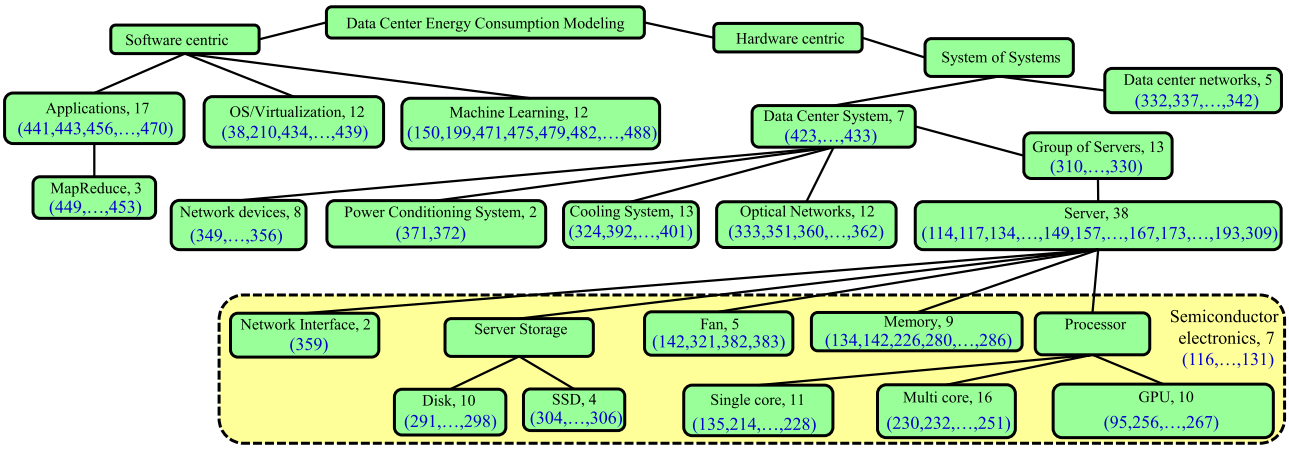


Fig. 34. Taxonomy of power consumption models. The numbers between the parentheses correspond to the reference in this paper. The number following the title of each box indicates the total number of power consumption models described in this paper from this category. Note that the number of references do not necessarily relate to the number of power models shown here.

prediction. Artificial neural network (ANN) and linear regression have been used to develop prediction-based resource measurement and provisioning strategies to satisfy future resource demands [379]. In this work Islam *et al.* used data generated from the TPC-W benchmark in the Amazon EC2 cloud for training and testing the prediction models. They validated the effectiveness of the prediction framework and claimed that their framework is able to make accurate projections of energy consumption requirement and can also forecast resource demand before the VM instance's setup time.

XI. Comparison of Techniques for Energy Consumption Modeling

The first half of this paper focused on power consumption modeling efforts made at various different levels of abstraction in data centers. A summary of how different power model equations map to different levels of a data center's components hierarchy is given in Fig. 34. Most of the power modeling efforts have been conducted for lower level hardware systems. In the literature surveyed, we observed that the majority of power models are developed around processor power consumption. There have been relatively few models for other important components such as system fans or SSDs. This may be partly due to the fact that most of the power modeling was carried out as part of a energy consumption reduction mechanism, which focus only on energy proportional systems. This may be another reason for why there are very few works on network level power modeling, since network devices are less power proportional compared to the servers.

A. Power Model Complexity

Relatively few power models currently exists for higher levels of the data center component hierarchy. Power modeling research on the OS/virtualization layer of data centers still lag behind the work on the physical hardware and application layers. There are a number of reasons for the lack of research, including the complexity of systems at higher levels of the data center hierarchy. A chronological summary of data center power modeling and prediction research is shown in Fig. 35. We see that the amount of power modeling and prediction research has increased significantly in the last two years time.

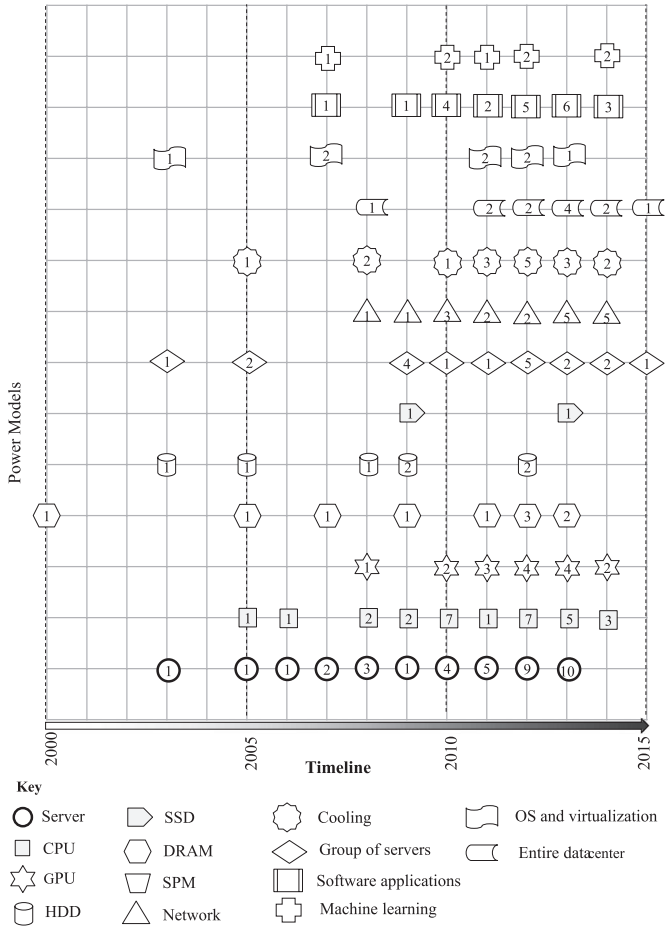


Fig. 35. Data center energy consumption modeling and prediction timeline. The number on each icon represents the number of publications pertaining to the power models of the icon's category. The timeline highlights both power consumption models and power consumption prediction techniques.

We observed that certain power models are built on top of the others. For example the power model described in Equation (24) describing the CPU power consumption has been extended to the entire system's power consumption in the power model described in the Equation (110). We observed that certain power models such as the one described in Equations (22) and (7) has been highly influential in power modeling at various levels of data centers. While there are works that assume the applications running in the cluster are bound to one type of resource such as CPU [380], power usage of individual hardware devices may not necessarily provide an overall view of the power consumption by the entire system [381]. For example, recent work have shown that memory intensive algorithm implementations may consume more energy than CPU intensive algorithms in the context of sorting and joining algorithms which are essential for database query processing [382], [383]. Simple models that work well in the context of hardware many not necessarily work well in the context of software, since there are many components handled by a software system running in a typical data center and these components change quite frequently (e.g., certain old/malfunctioning hardware components in a server many be replaced with new ones). Most of the hardware level power models do not rely on machine learning techniques while the work conducted at the OS/virtualization or application levels rely more heavily on machine learning.

Various new hardware technologies such as FPGAs show great potential in being deployed in future data centers [384]–[386]. Since such technologies are relatively new in data centers we do not describe power models for them in detail in this paper.

Most of the current power consumption models are component/application/system centric. Hence these models introduce multiple challenges when employed to model energy consumption of a data center. These challenges are listed below,

- *Portability*: The power consumption models are not portable across different systems.
- *Accuracy*: Since the workloads are diverse across different data center deployments, the accuracy of the models degrade rapidly. Furthermore, the level of support given by most modern hardware systems for measuring energy consumption is insufficient.

While some studies have solely attributed a server's power consumption to the CPU, recent studies have shown that processors contribute only a minority of many server's power demand. Furthermore, it has been found that the chipset has become the dominant source of power consumption in modern commodity servers [387], [388]. Therefore, in recent works there has been a tendency to use non-CPU-centric power models.

The granularity of power measurement and modeling is an important consideration. The appropriate measurement granularity depends on the application [128]. However, most of the available hardware infrastructures do not allow for power consumption monitoring of the individual applications. Therefore, the validity of the most power models with multiple applications running in a data center still needs to be evaluated. Furthermore, most of the power modeling research do not attempt to characterize the level of accuracy of their proposed power models, which makes it harder to do comparisons among them.

While there has been previous efforts to model the energy consumed by software [389], it is difficult to estimate the power consumption of real world data center software applications [127] due to their complexity. There have been a few notable efforts conducted to alleviate this issue, such as PowerPack [56], [70].

As stated earlier, it is possible to combine multiple power models corresponding to different components of a system to construct power models representing the entire system. This can go beyond the simple additive models as used in for example Equations (148) and (76). Certain works have used more sophisticated combinations of base models. For example, Roy *et al.* [85] combined the processor and memory power models described in Equations (54) and (87) respectively into the following power model as,

$$E(A) = W(A) + C(A)\frac{PB}{k}, \tag{210}$$

where there are P parallel disks, each block has B items, and the fast memory can hold P blocks. The number of Activation Cycles used by algorithm A is denoted by $C(A)$. Work complexity is defined by $W(A)$ and C is the ACT command.

B. Effectiveness of the Power Models

The effectiveness of the power models we have surveyed is another big question. Certain power models created for hardware systems have been demonstrated to be more effective than power models constructed for software systems. Furthermore, most of the current power models assume static system behavior. Although more than 200 power models have been surveyed in this paper, most of them are applicable at different levels of granularity of the data center system. Power models which have been used in real world application deployments are the most effective in this regard. Power model used in DVFS described in Equation (7), Wattch CPU power modeling framework [81], Joulemeter VM power metering [88] framework, McPAT processor power modeling framework are examples for such initiatives.

While some of the power models used in the lower level hardware systems rely on physical and electrical principles, the power consumption models at higher levels have a less tangible physical basis. For example, the popular power model in Equation (7) represents the physics of operation of lower level hardware and is more tangible than, for example, the neural network based power model described in [373]. Additionally, we observe that reinforcement learning has been used more extensively for power prediction than other machine learning techniques.

C. Applications of the Power Models

The power models described in this paper have a number of different applications. One of the key applications is

power consumption prediction. If a model is calibrated to a particular data center environment (i.e., system state ($\vec{S}$) and execution strategy ($\vec{E}$) in Equation (2)), and the relevant input parameters are known in advance (i.e., input to the application ($\vec{A}$) in Equation (2)), they can be used for predicting the target system or subsystem's energy consumption, as illustrated in the energy consumption modeling and prediction process shown in Fig. 2 in Section III.

Some of these applications can be found in the areas of energy consumption efficiency such as smart grids, sensor networks, content delivery networks, etc. Theoretical foundations of Smart Metering is similar to the power consumption modeling work described in this paper. Data center power models can be used in conjunction with smart meter networks to optimize the energy costs of data center systems [390].

Improving the impact of data center's load on power systems (due to the data center's massive energy usage) and reducing the cost of energy for data centers are two of the most important objectives in energy consumption optimization of data centers. The energy consumption models surveyed in this paper helps achieving the aforementioned objectives in multiple different ways when considering their applications in different power related studies such as electricity market participation, renewable power integration [318], [319], demand response, carbon market, etc. Based on the availability of power plants and fuels, local fuel costs, and pricing regulations electricity price exhibits location diversities as well as time diversities [292]. For example, the power model (shown in Equation (168)) presented for entire data center by Yao *et al.* has been used for reducing energy consumption by 18% through extensive simulation based experiments [121] on geographically distributed data centers.

XII Future Directions

The aim of this survey paper was to create a comprehensive taxonomy of data center power modeling and prediction techniques. A number of different insights gained through this survey were described in Section XI. It can be observed that most of the existing power modeling techniques do not consider the interactions occurring between different components of systems. For example, the power consumption analysis conducted by Zhao *et al.* indicates that DRAM access consumes a significant portion of the total system power. These kinds of intercomponent relationships should be considered when developing power consumption models [179].

Multiple future directions for energy consumption modeling and prediction exists in the areas of neural computing, bioinspired computing [391], [392], etc. There have been recent work on application of bioinspired techniques for development of novel data center architectures [393], [394].

Swarm Intelligence is an AI technique and a branch of evolutionary computing that works on the collective behavior of systems having many individuals interacting locally with each other and with their environment [395]. In recent times, Swarm-inspired techniques have been employed for reducing power consumption in data centers in [396]–[398].

Support vector machines (SVMs) are one of the most popular algorithms in machine learning. SVMs often provide significantly better classification performance than other machine learning algorithms on reasonably sized data sets. Currently SVM is popular among the researchers in electricity distribution systems, and they have applications such as power-quality classification, power transformer fault diagnosis, etc. However, not much work have been conducted on using SVMs for building power models.

Deep learning [399] is a novel field of research in AI that deals with deep architectures which are composed of multiple levels of non-linear operations. Examples for deep architectures include neural networks with many hidden layers, complicated propositional formulas re-using many sub-formulas [400], etc. Deep neural networks have been proven to produce exceptional results in classification tasks [401] which indicates its promise in creating future high-resolution energy models for data centers.

Data center operators are interested in using alternative power sources [402] (such as wind power, solar power [403], etc.) for data center operations, both for cost savings and as an environmentally friendly measure [404]. For example, it has been shown recently that UPS batteries used in data centers can help reduce the peak power costs without any workload performance degradation [405], [406]. Furthermore, direct current (DC) power distribution, battery-backed servers/racks [407] to improve on central UPS power backups, free cooling [408] (practice of using outside air to cool the data center facility), thermal energy storage [409], etc. are some trends in data center energy related research [410]. Modeling power consumption of such complex modes of data center operations is another prospective future research direction.

XIII. Summary

Data centers are the backbone of today's Internet and cloud computing systems. Due to the increasing demand for electrical energy by data centers, it is necessary to account for the vast amount of energy they consume. Energy modeling and prediction of data centers plays a pivotal role in this context.

This survey paper conducted a systematic, in-depth study about existing work in power consumption modeling for data center systems. We performed a layer-wise decomposition of a data center system's power hierarchy. We first divided the components into two types, hardware and software. Next, we conducted an analysis of current power models at different layers of the data center system in a bottom-up fashion. Altogether, more than 200 power models were examined in this survey.

We observed that while there has been a large number of studies conducted on the energy consumption modeling at lower levels of the data center hierarchy, much less work has been done at the higher levels [263]. This is a critical limitation of the current state-of-the-art in data center power modeling research. Furthermore, the accuracy, generality, and practicality of the majority of the power consumption models remain open. Based on the trend observed through our study we envision significant growth in energy modeling and prediction research for higher layers of data center systems in the near future.

REFERENCES

[1] R. Buyya, C. Vecchiola, and S. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*. Amsterdam, The Netherlands: Elsevier, 2013.

[2] R. Buyya, A. Beloglazov, and J. H. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," *CoRR*, vol. abs/1006.0308, 2010.

[3] L. Krug, M. Shackleton, and F. Saffre, "Understanding the environmental costs of fixed line networking," in *Proc. 5th Int. Conf. Future e-Energy Syst.*, 2014, pp. 87–95.

[4] Intel Intelligent Power Technology, Canalys, Shanghai, China, 2012. [Online]. Available: http://www.canalys.com/newsroom/data-center-infrastructure-market-will-be-worth-152-billion-2016

[5] M. Poess and R. O. Nambiar, "Energy cost, the key challenge of today's data centers: A power consumption analysis of TPC-C results," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1229–1240, Aug. 2008.

[6] Y. Gao, H. Guan, Z. Qi, B. Wang, and L. Liu, "Quality of service aware power management for virtualized data centers," *J. Syst. Architect.*, vol. 59, no. 4/5, pp. 245–259, Apr./May 2013.

[7] S. Rivoire, M. Shah, P. Ranganathan, C. Kozyrakis, and J. Meza, "Models and metrics to enable energy-efficiency optimizations," *Computer*, vol. 40, no. 12, pp. 39–48, Dec. 2007.

[8] K. Bilal, S. Malik, S. Khan, and A. Zomaya, "Trends and challenges in cloud datacenters," *IEEE Cloud Comput.*, vol. 1, no. 1, pp. 10–20, May 2014.

[9] B. Whitehead, D. Andrews, A. Shah, and G. Maidment, "Assessing the environmental impact of data centres—Part 1: Background, energy use and metrics," *Building Environ.*, vol. 82, pp. 151–159, Dec. 2014.

[10] V. Mathew, R. K. Sitaraman, and P. J. Shenoy, "Energy-aware load balancing in content delivery networks," *CoRR*, vol. abs/1109.5641, 2011.

[11] P. Corcoran and A. Andrae, "Emerging trends in electricity consumption for consumer ICT," Nat. Univ. Ireland, Galway, Ireland, Tech. Rep., 2013.

[12] J. Koomey, *Growth in Data Center Electricity Use 2005 to 2010*. Burlingame, CA, USA: Analytics Press, 2011. [Online]. Available: http://www.analyticspress.com/datacenters.html

[13] W. V. Heddeghem *et al.*, "Trends in worldwide ICT electricity consumption from 2007 to 2012," *Comput. Commun.*, vol. 50, pp. 64–76, Sep. 2014.

[14] "Energy efficiency policy options for Australian and New Zealand data centres," The Equipment Energy Efficiency (E3) Program, 2014.

[15] Info-Tech, "Top 10 energy-saving tips for a greener data center," Info-Tech Research Group, London, ON, Canada, Apr. 2010. [Online]. Available: http://static.infotech.com/downloads/samples/070411_premium_oo_greendc_top_10.pdf

[16] S. Yeo, M. M. Hossain, J.-C. Huang, and H.-H. S. Lee, "ATAC: Ambient temperature-aware capping for power efficient datacenters," in *Proc. ACM SOCC*, 2014, pp. 17:1–17:14.

[17] D. J. Brown and C. Reams, "Toward energy-efficient computing," *Queue*, vol. 8, no. 2, pp. 30:30–30:43, Feb. 2010.

[18] F. Bellosa, "The benefits of event: Driven energy accounting in power-sensitive systems," in *Proc. 9th Workshop ACM SIGOPS EQ—Beyond PC—New Challenges Oper. Syst.*, 2000, pp. 37–42.

[19] S.-Y. Jing, S. Ali, K. She, and Y. Zhong, "State-of-the-art research study for green cloud computing," *J. Supercomput.*, vol. 65, no. 1, pp. 445–468, Jul. 2013.

[20] Y. Hotta *et al.*, "Profile-based optimization of power performance by using dynamic voltage scaling on a PC cluster," in *Proc. 20th IPDPS*, Apr. 2006, pp. 1–8.

[21] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," in *Proc. 1st USENIX Conf. OSDI*, 1994, pp. 1–11.

[22] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proc. 20th Int. Symp. HPDC*, 2011, pp. 171–182.

[23] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proc. 10th IEEE/ACM Int. CCGrid*, May 2010, pp. 826–831.

[24] E. Feller, C. Rohr, D. Margery, and C. Morin, "Energy management in iaas clouds: A holistic approach," in *Proc. IEEE 5th Int. Conf. CLOUD Comput.*, Jun. 2012, pp. 204–212.

[25] C. Lefurgy, X. Wang, and M. Ware, "Power capping: A prelude to power shifting," *Cluster Comput.*, vol. 11, no. 2, pp. 183–195, Jun. 2008.

[26] M. Lin, A. Wierman, L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013.

[27] N. Seel, Ed., *Mathematical Models*, Encyclopedia of the Sciences of Learning. New York, NY, USA: Springer-Verlag, 2012, p. 2113.

[28] S. M. Rivoire, "Models and metrics for energy-efficient computer systems," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2008.

[29] M. vor dem Berge *et al.*, "Modeling and simulation of data center energy-efficiency in coolemall," in *Energy Efficient Data Centers*, vol. 7396, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2012, pp. 25–36.

[30] A. Floratou, F. Bertsch, J. M. Patel, and G. Laskaris, "Towards building wind tunnels for data center design," *Proc. VLDB Endowment*, vol. 7, no. 9, pp. 781–784, May 2014.

[31] D. Kilper *et al.*, "Power trends in communication networks," *IEEE J. Sel. Topics Quantum Electron.*, vol. 17, no. 2, pp. 275–284, Mar. 2011.

[32] H. Xu and B. Li, "Reducing electricity demand charge for data centers with partial execution," in *Proc. 5th Int. Conf. Future e-Energy Syst.*, 2014, pp. 51–61.

[33] D. Wang *et al.*, "ACE: Abstracting, characterizing and exploiting data-center power demands," in *Proc. IEEE IISWC*, Sep. 2013, pp. 44–55.

[34] J. Evans, "On performance and energy management in high performance computing systems," in *Proc. 39th ICPPW*, Sep. 2010, pp. 445–452.

[35] V. Venkatachalam and M. Franz, "Power reduction techniques for microprocessor systems," *ACM Comput. Surveys*, vol. 37, no. 3, pp. 195–237, Sep. 2005.

[36] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Adv. Comput.*, vol. 82, no. 11, pp. 47–111, 2011.

[37] J. Wang, L. Feng, W. Xue, and Z. Song, "A survey on energy-efficient data management," *SIGMOD Rec.*, vol. 40, no. 2, pp. 17–23, Sep. 2011.

[38] S. Reda and A. N. Nowroz, "Power modeling and characterization of computing devices: A survey," *Found. Trends Electron. Des. Autom.*, vol. 6, no. 2, pp. 121–216, Feb. 2012.

[39] C. Ge, Z. Sun, and N. Wang, "A survey of power-saving techniques on data centers and content delivery networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1334–1354, 3rd Quart. 2013.

[40] T. Bostoen, S. Mullender, and Y. Berbers, "Power-reduction techniques for data-center storage systems," *ACM Comput. Surveys*, vol. 45, no. 3, pp. 33:1–33:38, Jul. 2013.

[41] A.-C. Orgerie, M. D. de Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Comput. Surveys*, vol. 46, no. 4, pp. 47:1–47:31, Mar. 2014.

[42] S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems," *arXiv preprint arXiv:1401.0765*, 2014.

[43] S. Mittal and J. S. Vetter, "A survey of methods for analyzing and improving GPU energy efficiency," *ACM Comput. Surveys*, vol. 47, no. 2, pp. 19:1–19:23, Aug. 2014.

[44] A. Hammadi and L. Mhamdi, "A survey on architectures and energy efficiency in data center networks," *Comput. Commun.*, vol. 40, pp. 1–21, Mar. 2014.

[45] K. Bilal *et al.*, "A taxonomy and survey on green data center networks," *Future Gener. Comput. Syst.*, vol. 36, pp. 189–208, Jul. 2014.

[46] K. Bilal, S. Khan, and A. Zomaya, "Green data center networks: Challenges and opportunities," in *Proc. 11th Int. Conf. FIT*, Dec. 2013, pp. 229–234.

[47] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renew. Sustain. Energy Rev.*, vol. 31, pp. 622–638, Mar. 2014.

[48] A. Rahman, X. Liu, and F. Kong, "A survey on geographic load balancing based data center power management in the smart grid environment," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 214–233, 1st Quart. 2014.

[49] S. Mittal, "Power management techniques for data centers: A survey," *CoRR*, vol. abs/1404.6681, 2014.

[50] C. Gu, H. Huang, and X. Jia, "Power metering for virtual machine in cloud computing-challenges and opportunities," *IEEE Access*, vol. 2, pp. 1106–1116, Sep. 2014.

[51] F. Kong and X. Liu, "A survey on green-energy-aware power management for datacenters," *ACM Comput. Surveys*, vol. 47, no. 2, pp. 30:1–30:38, Nov. 2014.

[52] J. Shuja *et al.*, "Survey of techniques and architectures for designing energy-efficient data centers," *IEEE Syst. J.*, to be published.

[53] B. Khargharia *et al.*, "Autonomic power performance management for large-scale data centers," in *Proc. IEEE IPDPS*, Mar. 2007, pp. 1–8.

[54] Y. Joshi and P. Kumar, "Introduction to data center energy flow and thermal management," in *Energy Efficient Thermal Management of Data Centers*, Y. Joshi and P. Kumar, Eds. New York, NY, USA: Springer-Verlag, 2012, pp. 1–38.

[55] Y. Kodama *et al.*, "Power reduction scheme of fans in a blade system by considering the imbalance of CPU temperatures," in *Proc. IEEE/ACM Int. Conf. CPSCom GreenCom*, Dec. 2010, pp. 81–87.

[56] R. Ge *et al.*, "Powerpack: Energy profiling and analysis of high-performance systems and applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 658–671, May 2010.

[57] H. Nagasaka, N. Maruyama, A. Nukada, T. Endo, and S. Matsuoka, "Statistical power modeling of GPU kernels using performance counters," in *Proc. Int. Green Comput. Conf.*, Aug. 2010, pp. 115–122.

[58] W. Bircher and L. John, "Core-level activity prediction for multicore power management," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 1, no. 3, pp. 218–227, Sep. 2011.

[59] S. Song, C. Su, B. Rountree, and K. Cameron, "A simplified and accurate model of power-performance efficiency on emergent GPU architectures," in *Proc. IEEE 27th IPDPS*, May 2013, pp. 673–686.

[60] J. Smith, A. Khajeh-Hosseini, J. Ward, and I. Sommerville, "Cloudmonitor: Profiling power usage," in *Proc. IEEE 5th CLOUD Comput.*, Jun. 2012, pp. 947–948.

[61] M. Witkowski, A. Oleksiak, T. Piontek, and J. Weglarz, "Practical power consumption estimation for real life {HPC} applications," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 208–217, Jan. 2013.

[62] Intelligent Rack Power Distribution, Raritan, Somerset, NJ, USA, 2010.

[63] R. Bolla, R. Bruschi, and P. Lago, "The hidden cost of network low power idle," in *Proc. IEEE ICC*, Jun. 2013, pp. 4148–4153.

[64] M. Burtscher, I. Zecena, and Z. Zong, "Measuring GPU power with the k20 built-in sensor," in *Proc. Workshop GPGPU-7*, 2014, pp. 28:28–28:36.

[65] S. Miwa and C. R. Lefurgy, "Evaluation of core hopping on POWER7," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 55–60, Dec. 2014.

[66] HP, Server Remote Management With hp Integrated Lights Out (ILO), 2014. [Online]. Available: http://h18013.www1.hp.com/products/servers/management/remotemgmt.html

[67] H. Chen and W. Shi, "Power measuring and profiling: The state of art," in *Handbook of Energy-Aware and Green Computing*. Boca Raton, FL, USA: CRC Press, 2012, pp. 649–674.

[68] L. John and L. Eeckhout, *Performance Evaluation and Benchmarking*. New York, NY, USA: Taylor & Francis, 2005.

[69] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proc. 36th Annu. IEEE/ACM Int. Symp. MICRO*, 2003, pp. 93.

[70] X. Wu *et al.*, "MuMMi: Multiple metrics modeling infrastructure for exploring performance and power modeling," in *Proc. Conf. XSEDE—Gateway Discov.*, 2013, pp. 36:1–36:8.

[71] J. Shuja, K. Bilal, S. Madani, and S. Khan, "Data center energy efficient resource scheduling," *Cluster Comput.*, vol. 17, no. 4, pp. 1265–1277, Dec. 2014.

[72] Y. Chen *et al.*, "Managing server energy and operational costs in hosting centers," in *Proc. of the 2005 ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2005, pp. 303–314.

[73] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A comparison of high-level full-system power models," in *Proc. Conf. HotPower Aware Comput. Syst.*, 2008, p. 3.

[74] G.-Y. Wei, M. Horowitz, and J. Kim, "Energy-efficient design of high-speed links," in *Power Aware Design Methodologies*, M. Pedram and J. Rabaey, Eds. New York, NY, USA: Springer-Verlag, 2002, pp. 201–239.

[75] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Y. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *CoRR*, 2010.

[76] T. Burd and R. Brodersen, *Energy Efficient Microprocessor Design*. New York, NY, USA: Springer-Verlag, 2002.

[77] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, "Efficient power modeling and software thermal sensing for runtime temperature monitoring," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 12, no. 3, pp. 25:1–25:29, May 2008.

[78] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, ser. The Morgan Kaufmann Series in Computer Architecture and Design. San Mateo, CA, USA: Morgan Kaufmann, 2011.

[79] Y. C. Lee and A. Zomaya, "Energy conscious scheduling for distributed computing systems under different operating conditions," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1374–1381, Aug. 2011.

[80] R. Ge, X. Feng, and K. W. Cameron, "Performance-constrained distributed DVS scheduling for scientific applications on power-aware clusters," in *Proc. ACM/IEEE Conf. SC*, 2005, p. 34.

[81] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. 27th Annu. ISCA*, 2000, pp. 83–94.

[82] S. Yeo and H.-H. Lee, "Peeling the power onion of data centers," in *Energy Efficient Thermal Management of Data Centers*, Y. Joshi and P. Kumar, Eds. New York, NY, USA: Springer-Verlag, 2012, pp. 137–168.

[83] J. Esch, "Prolog to 'Estimating the energy use and efficiency potential of U.S. Data Centers," *Proc. IEEE*, vol. 99, no. 8, pp. 1437–1439, Aug. 2011.

[84] S. Ghosh, S. Chandrasekaran, and B. Chapman, "Statistical modeling of power/energy of scientific kernels on a multi-gpu system," in *Proc. IGCC*, Jun. 2013, pp. 1–6.

[85] S. Roy, A. Rudra, and A. Verma, "An energy complexity model for algorithms," in *Proc. 4th Conf. ITCS*, 2013, pp. 283–304.

[86] R. Jain, D. Molnar, and Z. Ramzan, "Towards understanding algorithmic factors affecting energy consumption: Switching complexity, randomness, and preliminary experiments," in *Proc. Joint Workshop Found. Mobile Comput. DIALM-POMC*, 2005, pp. 70–79.

[87] B. M. Tudor and Y. M. Teo, "On understanding the energy consumption of arm-based multicore servers," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 1, pp. 267–278, Jun. 2013.

[88] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proc. 1st ACM SoCC*, 2010, pp. 39–50.

[89] R. Ge, X. Feng, and K. Cameron, "Modeling and evaluating energy-performance efficiency of parallel processing on multicore based power aware systems," in *Proc. IEEE IPDPS*, May 2009, pp. 1–8.

[90] S. L. Song, K. Barker, and D. Kerbyson, "Unified performance and power modeling of scientific workloads," in *Proc. 1st Int. Workshop E2SC*, 2013, pp. 4:1–4:8.

[91] A. Lewis, J. Simon, and N.-F. Tzeng, "Chaotic attractor prediction for server run-time energy consumption," in *Proc. Int. Conf. HotPower Aware Comput. Syst.*, 2010, pp. 1–16.

[92] A. W. Lewis, N.-F. Tzeng, and S. Ghosh, "Runtime energy consumption estimation for server workloads based on chaotic time-series approximation," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 3, pp. 15:1–15:26, Oct. 2012.

[93] V. Perumal and S. Subbiah, "Power-conservative server consolidation based resource management in cloud," *Int. J. Netw. Manage.*, vol. 24, no. 6, pp. 415–432, Nov./Dec. 2014.

[94] A. Chatzipapas *et al.*, "Challenge: Resolving data center power bill disputes: The energy-performance trade-offs of consolidation," in *Proc. ACM 6th Int. Conf. Future e-Energy Syst.*, 2015, pp. 89–94. [Online]. Available: http://doi.acm.org/10.1145/2768510.2770933

[95] I. Alan, E. Arslan, and T. Kosar, "Energy-aware data transfer tuning," in *Proc. 14th IEEE/ACM Int. Symp. CCGrid*, May 2014, pp. 626–634.

[96] A. Lewis, S. Ghosh, and N.-F. Tzeng, "Run-time energy consumption estimation based on workload in server systems," in *Proc. Conf. HotPower Aware Comput. Syst.*, 2008, p. 4.

[97] P. Bohrer *et al.*, "The case for power management in web servers," in *Power Aware Computing*, R. Graybill and R. Melhem, Eds. Norwell, MA, USA: Kluwer, 2002, pp. 261–289.

[98] R. Lent, "A model for network server performance and power consumption," *Sustainable Comput., Informat. Syst.*, vol. 3, no. 2, pp. 80–93, Jun. 2013.

[99] F. Chen, J. Grundy, Y. Yang, J.-G. Schneider, and Q. He, "Experimental analysis of task-based energy consumption in cloud computing systems," in *Proc. 4th ACM/SPEC ICPE*, 2013, pp. 295–306.

[100] P. Xiao, Z. Hu, D. Liu, G. Yan, and X. Qu, "Virtual machine power measuring technique with bounded error in cloud environments," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 818–828, Mar. 2013.

[101] G. Dhiman, K. Mihic, and T. Rosing, "A system for online power prediction in virtualized environments using Gaussian mixture models," in *Proc. 47th DAC*, 2010, pp. 807–812.

[102] L. A. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," in *Synthesis Lectures on Computer Architecture*, 1st ed., vol. 4. San Rafel, CA, USA: Morgan & Claypool, 2009, pp. 1–108.

[103] K. T. Malladi *et al.*, "Towards energy-proportional datacenter memory with mobile DRAM," in *Proc. 39th Annu. ISCA*, 2012, pp. 37–48.

[104] D. Economou, S. Rivoire, and C. Kozyrakis, "Full-system power analysis and modeling for server environments," in *Proc. Workshop MOBS*, 2006, pp. 70–77.

[105] A.-C. Orgerie, L. Lefèvre, and I. Guérin-Lassous, "Energy-efficient bandwidth reservation for bulk data transfers in dedicated wired networks," *J. Supercomput.*, vol. 62, no. 3, pp. 1139–1166, Dec. 2012.

[106] Y. Li, Y. Wang, B. Yin, and L. Guan, "An online power metering model for cloud environment," in *Proc. 11th IEEE Int. Symp. NCA*, Aug. 2012, pp. 175–180.

[107] X. Xu, K. Teramoto, A. Morales, and H. Huang, "Dual: Reliability-aware power management in data centers," in *Proc. 13th IEEE/ACM Int. Symp. CCGrid*, May 2013, pp. 530–537.

[108] E. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," in *Power-Aware Computer Systems*, vol. 2325, ser. Lecture Notes in Computer Science, B. Falsafi and T. Vijaykumar, Eds. Berlin, Germany: Springer-Verlag, 2003, pp. 179–197.

[109] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. 34th Annu. ISCA*, 2007, pp. 13–23.

[110] Y. Jin, Y. Wen, Q. Chen, and Z. Zhu, "An empirical investigation of the impact of server virtualization on energy efficiency for green data center," *Comput. J.*, vol. 56, no. 8, pp. 977–990, Aug. 2013.

[111] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.

[112] R. Basmadjian, N. Ali, F. Niedermeier, H. de Meer, and G. Giuliani, "A methodology to predict the power consumption of servers in data centres," in *Proc. 2nd Int. Conf. e-Energy-Efficient Comput. Netw.*, 2011, pp. 1–10.

[113] V. Gupta, R. Nathuji, and K. Schwan, "An analysis of power reduction in datacenters using heterogeneous chip multiprocessors," *SIGMETRICS Perform. Eval. Rev.*, vol. 39, no. 3, pp. 87–91, Dec. 2011.

[114] X. Zhang, J.-J. Lu, X. Qin, and X.-N. Zhao, "A high-level energy consumption model for heterogeneous data centers," *Simul. Model. Pract. Theory*, vol. 39, pp. 41–55, Dec. 2013.

[115] M. Tang and S. Pan, "A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers," *Neural Process. Lett.*, vol. 41, no. 2, pp. 1–11, Apr. 2014.

[116] S. ul Islam and J.-M. Pierson, "Evaluating energy consumption in CDN servers," in *ICT as Key Technology Against Global Warming*, vol. 7453, ser. Lecture Notes in Computer Science, A. Auweter, D. Kranzlmüller, A. Tahamtan, and A. Tjoa, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 64–78.

[117] S.-H. Lim, B. Sharma, B. C. Tak, and C. Das, "A dynamic energy management scheme for multi-tier data centers," in *Proc. IEEE ISPASS*, 2011, pp. 257–266.

[118] Z. Wang, N. Tolia, and C. Bash, "Opportunities and challenges to unify workload, power, and cooling management in data centers," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 3, pp. 41–46, Aug. 2010.

[119] H. Li, G. Casale, and T. Ellahi, "SLA-driven planning and optimization of enterprise applications," in *Proc. 1st Joint WOSP/SIPEW Int. Conf. Perform. Eng.*, 2010, pp. 117–128.

[120] C.-J. Tang and M.-R. Dai, "Dynamic computing resource adjustment for enhancing energy efficiency of cloud service data centers," in *Proc. IEEE/SICE Int. Symp. SII*, Dec. 2011, pp. 1159–1164.

[121] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1431–1439.

[122] Y. Tian, C. Lin, and K. Li, "Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing," *Cluster Comput.*, vol. 17, no. 3, pp. 943–955, Sep. 2014.

[123] B. Subramaniam and W.-C. Feng, "Enabling efficient power provisioning for enterprise applications," in *Proc. IEEE/ACM 14th Int. Symp. CCGrid*, May 2014, pp. 71–80.

[124] S.-W. Ham, M.-H. Kim, B.-N. Choi, and J.-W. Jeong, "Simplified server model to simulate data center cooling energy consumption," *Energy Buildings*, vol. 86, pp. 328–339, Jan. 2015.

[125] T. Horvath and K. Skadron, "Multi-mode energy management for multi-tier server clusters," in *Proc. 17th Int. Conf. PACT*, 2008, pp. 270–279.

[126] C. Lefurgy, X. Wang, and M. Ware, "Server-level power control," in *Proc. IEEE 4th ICAC*, 2007, pp. 1–4.

[127] G. Da Costa and H. Hlavacs, "Methodology of measurement for energy consumption of applications," in *Proc. IEEE/ACM Int. Conf. GRID Comput.*, Oct. 2010, pp. 290–297.

[128] J. C. McCullough *et al.*, "Evaluating the effectiveness of model-based power characterization," in *Proc. USENIXATC*, 2011, p. 12.

[129] T. Enokido and M. Takizawa, "An extended power consumption model for distributed applications," in *Proc. IEEE 26th Int. Conf. AINA*, Mar. 2012, pp. 912–919.

[130] B. Mills, T. Znati, R. Melhem, K. Ferreira, and R. Grant, "Energy consumption of resilience mechanisms in large scale systems," in *Proc. 22nd Euromicro Int. Conf. PDP*, Feb. 2014, pp. 528–535.

[131] M. Pawlish, A. Varde, and S. Robila, "Analyzing utilization rates in data centers for optimizing energy management," in *Proc. IGCC*, Jun. 2012, pp. 1–6.

[132] V. Maccio and D. Down, "On optimal policies for energy-aware servers," in *Proc. IEEE 21st Int, Symp. MASCOTS*, Aug. 2013, pp. 31–39.

[133] Q. Deng, L. Ramos, R. Bianchini, D. Meisner, and T. Wenisch, "Active low-power modes for main memory with memscale," *IEEE Micro*, vol. 32, no. 3, pp. 60–69, May 2012.

[134] Q. Deng, D. Meisner, A. Bhattacharjee, T. F. Wenisch, and R. Bianchini, "Multiscale: Memory system dvfs with multiple memory controllers," in *Proc. ACM/IEEE ISLPED*, 2012, pp. 297–302.

[135] Q. Deng, D. Meisner, A. Bhattacharjee, T. F. Wenisch, and R. Bianchini, "Coscale: Coordinating CPU and memory system dvfs in server systems," in *Proc. IEEE/ACM 45th Annu. Int. Symp. MICRO*, 2012, pp. 143–154.

[136] J. Janzen, "Calculating memory system power for DDR SDRAM," *Micron Designline*, vol. 10, no. 2, pp. 1–12, 2nd Quart. 2001.

[137] Calculating Memory System Power for DDR3, Micron Technology, Inc., Boise, ID, USA, 2007.

[138] J. Jeffers and J. Reinders, "Chapter 1–Introduction," in *Intel Xeon Phi Coprocessor High Performance Programming*, J. Jeffers and J. Reinders, Eds. Boston, MA, USA: Morgan Kaufmann, 2013, pp. 1–22.

[139] The Problem of Power Consumption in Servers, Energy Efficiency for Information Technology, Intel, Santa Clara, CA, USA, 2009.

[140] I. Moreno and J. Xu, "Neural network-based overallocation for improved energy-efficiency in real-time cloud environments," in *Proc. IEEE 15th ISORC*, Apr. 2012, pp. 119–126.

[141] R. Joseph and M. Martonosi, "Run-time power estimation in high performance microprocessors," in *Proc. Int. Symp. Low Power Electron. Des.*, 2001, pp. 135–140.

[142] R. Hameed *et al.*, "Understanding sources of inefficiency in general-purpose chips," in *Proc. ISCA*, 2010, pp. 37–47.

[143] S. Li *et al.*, "The MCPAT framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 1, pp. 5:1–5:29, Apr. 2013.

[144] V. Zyuban *et al.*, "Power optimization methodology for the IBM POWER7 microprocessor," *IBM J. Res. Develop.*, vol. 55, no. 3, pp. 7:1–7:9, May 2011.

[145] T. Diop, N. E. Jerger, and J. Anderson, "Power modeling for heterogeneous processors," in *Proc. Workshop GPGPU-7*, 2014, pp. 90:90–90:98.

[146] T. Li and L. K. John, "Run-time modeling and estimation of operating system power consumption," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2003, pp. 160–171.

[147] M. Jarus, A. Oleksiak, T. Piontek, and J. Weglarz, "Runtime power usage estimation of HPC servers for various classes of real-life applications," *Future Gener. Comput. Syst.*, vol. 36, pp. 299–310, Jul. 2014.

[148] D. Shin *et al.*, "Energy-optimal dynamic thermal management for green computing," in *Proc. IEEE/ACM ICCAD*, 2009, pp. 652–657.

[149] R. Bertran, M. Gonzelez, X. Martorell, N. Navarro, and E. Ayguade, "A systematic methodology to generate decomposable and responsive power models for CMPS," *IEEE Trans. Comput.*, vol. 62, no. 7, pp. 1289–1302, Jul. 2013.

[150] W. Bircher and L. John, "Complete system power estimation: A trickle-down approach based on performance events," in *Proc. IEEE ISPASS*, 2007, pp. 158–168.

[151] R. Bertran, M. Gonzàlez, X. Martorell, N. Navarro, and E. Ayguadé, "Counter-based power modeling methods," *Comput. J.*, vol. 56, no. 2, pp. 198–213, Feb. 2013.

[152] A. Merkel, J. Stoess, and F. Bellosa, "Resource-conscious scheduling for energy efficiency on multicore processors," in *Proc. 5th EuroSys Conf. Comput.*, 2010, pp. 153–166.

[153] S. Wang, H. Chen, and W. Shi, "SPAN: A software power analyzer for multicore computer systems," *Sustainable Comput., Informat. Syst.*, vol. 1, no. 1, pp. 23–34, Mar. 2011.

[154] S. Hong and H. Kim, "An integrated GPU power and performance model," *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 280–289, Jun. 2010.

[155] W.-T. Shiue and C. Chakrabarti, "Memory exploration for low power, embedded systems," in *Proc. 36th Des. Autom. Conf.*, 1999, pp. 140–145.

[156] G. Contreras and M. Martonosi, "Power prediction for intel XScale processors using performance monitoring unit events," in *Proc. ISLPED*, 2005, pp. 221–226.

[157] X. Chen, C. Xu, and R. Dick, "Memory access aware on-line voltage control for performance and energy optimization," in *Proc. IEEE/ACM ICCAD*, Nov. 2010, pp. 365–372.

[158] A. Merkel and F. Bellosa, "Balancing power consumption in multiprocessor systems," in *Proc. 1st ACM SIGOPS/EuroSys Conf. Comput.*, 2006, pp. 403–414.

[159] R. Basmadjian and H. de Meer, "Evaluating and modeling power consumption of multi-core processors," in *Proc. 3rd Int. Conf. Future e-Energy Syst.—Where Energy, Comput. Commun. Meet*, 2012, pp. 12:1–12:10.

[160] K. Li, "Optimal configuration of a multicore server processor for managing the power and performance tradeoff," *J. Supercomput.*, vol. 61, no. 1, pp. 189–214, Jul. 2012.

[161] J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 45–58, Jan. 2014.

[162] Q. Liu *et al.*, "Hardware support for accurate per-task energy metering in multicore systems," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 4, pp. 34:1–34:27, Dec. 2013.

[163] W. Shi, S. Wang, and B. Luo, "CPT: An energy-efficiency model for multi-core computer systems," in *Proc. 5th Workshop Energy-Efficient Des.*, 2013, pp. 1–6.

[164] O. Sarood, A. Langer, A. Gupta, and L. Kale, "Maximizing throughput of overprovisioned HPC data centers under a strict power budget," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. SC*, 2014, pp. 807–818.

[165] V. Jiménez *et al.*, "Power and thermal characterization of power6 system," in *Proc. 19th Int. Conf. PACT*, 2010, pp. 7–18.

[166] R. Bertran, A. Buyuktosunoglu, M. Gupta, M. Gonzalez, and P. Bose, "Systematic energy characterization of CMP/SMT processor systems via automated micro-benchmarks," in *Proc. IEEE/ACM 45th Int. Symp. MICRO*, Dec. 2012, pp. 199–211.

[167] R. Bertran *et al.*, "Accurate energy accounting for shared virtualized environments using pmc-based power modeling techniques," in *Proc. IEEE/ACM 11th Int. Conf. GRID Comput.*, 2010, pp. 1–8.

[168] R. Bertran, M. Gonzàlez, X. Martorell, N. Navarro, and E. Ayguadé, "Counter-based power modeling methods: Top-down vs. bottom-up," *Comput. J.*, vol. 56, no. 2, pp. 198–213, Feb. 2012.

[169] X. Fu and X. Wang, "Utilization-controlled task consolidation for power optimization in multi-core real-time systems," in *Proc. IEEE 17th Int. Conf. Embedded RTCSA*, Aug. 2011, vol. 1, pp. 73–82.

[170] X. Qi and D. Zhu, "Power management for real-time embedded systems on block-partitioned multicore platforms," in *Proc. ICESS*, Jul. 2008, pp. 110–117.

[171] Y. Shao and D. Brooks, "Energy characterization and instruction-level energy model of Intel's Xeon Phi processor," in *Proc. IEEE ISLPED*, Sep. 2013, pp. 389–394.

[172] S. Kim, I. Roy, and V. Talwar, "Evaluating integrated graphics processors for data center workloads," in *Proc. Workshop HotPower-Aware Comput. Syst.*, 2013, pp. 8:1–8:5.

[173] J. Chen, B. Li, Y. Zhang, L. Peng, and J.-K. Peir, "Tree structured analysis on GPU power study," in *Proc. IEEE 29th ICCD*, Oct. 2011, pp. 57–64.

[174] J. Chen, B. Li, Y. Zhang, L. Peng, and J.-K. Peir, "Statistical GPU power analysis using tree-based methods," in *Proc. IGCC Workshops*, Jul. 2011, pp. 1–6.

[175] J. Leng *et al.*, "GPUWattch: Enabling energy optimizations in GPGPUs," *SIGARCH Comput. Archit. News*, vol. 41, no. 3, pp. 487–498, Jun. 2013.

[176] J. Leng, Y. Zu, M. Rhu, M. Gupta, and V. J. Reddi, "GPUVolt: Modeling and characterizing voltage noise in GPU architectures," in *Proc. ISLPED*, 2014, pp. 141–146.

[177] J. Lim *et al.*, "Power modeling for GPU architectures using MCPAT," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 19, no. 3, pp. 26:1–26:24, Jun. 2014.

[178] K. Kasichayanula *et al.*, "Power aware computing on GPUS," in *Proc. SAAHPC*, Jul. 2012, pp. 64–73.

[179] J. Zhao, G. Sun, G. H. Loh, and Y. Xie, "Optimizing GPU energy efficiency with 3D die-stacking graphics memory and reconfigurable memory interface," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 4, pp. 24:1–24:25, Dec. 2013.

[180] D.-Q. Ren and R. Suda, "Global optimization model on power efficiency of GPU and multicore processing element for SIMD computing with CUDA," *Comput. Sci. Res. Develop.*, vol. 27, no. 4, pp. 319–327, Nov. 2012.

[181] A. Marowka, "Analytical modeling of energy efficiency in heterogeneous processors," *Comput. Elect. Eng.*, vol. 39, no. 8, pp. 2566–2578, Nov. 2013.

[182] M. Rofouei, T. Stathopoulos, S. Ryffel, W. Kaiser, and M. Sarrafzadeh, "Energy-aware high performance computing with graphic processing units," in *Proc. HotPower Aware Comput. Syst. Conf.*, 2008, pp. 11–11.

[183] B. Giridhar *et al.*, "Exploring DRAM organizations for energy-efficient and resilient exascale memories," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. SC*, 2013, pp. 23:1–23:12.

[184] J. Lin *et al.*, "Software thermal management of DRAM memory for multicore systems," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2008, pp. 337–348.

[185] W. Stallings, *Computer Organization and Architecture: Designing for Performance*. London, U.K.: Pearson Education Inc., 2010.

[186] J. Lin, H. Zheng, Z. Zhu, H. David, and Z. Zhang, "Thermal modeling and management of DRAM memory systems," *SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 312–322, Jun. 2007.

[187] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye, "Energy-driven integrated hardware-software optimizations using simplepower," *SIGARCH Comput. Archit. News*, vol. 28, no. 2, pp. 95–106, May 2000.

[188] J. H. Ahn, N. P. Jouppi, C. Kozyrakis, J. Leverich, and R. S. Schreiber, "Improving system energy efficiency with memory rank subsetting," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 1, pp. 4:1–4:28, Mar. 2012.

[189] M. Rhu, M. Sullivan, J. Leng, and M. Erez, "A locality-aware memory hierarchy for energy-efficient GPU architectures," in *Proc. IEEE/ACM 46th Annu. Int. Symp. MICRO*, 2013, pp. 86–98.

[190] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu, "Memory power management via dynamic voltage/frequency scaling," in *Proc. 8th ACM ICAC*, 2011, pp. 31–40.

[191] K. T. Malladi *et al.*, "Rethinking DRAM power modes for energy proportionality," in *Proc. IEEE/ACM 45th Annu. Int. Symp. MICRO*, 2012, pp. 131–142.

[192] M. Sri-Jayantha, "Trends in mobile storage design," in *Proc. IEEE Symp. Low Power Electron.*, Oct. 1995, pp. 54–57.

[193] Y. Zhang, S. Gurumurthi, and M. R. Stan, "SODA: Sensitivity based optimization of disk architecture," in *Proc. 44th Annu. DAC*, 2007, pp. 865–870.

[194] S. Gurumurthi and A. Sivasubramaniam, *Energy-Efficient Storage Systems for Data Centers*. Hoboken, NJ, USA: Wiley, 2012, pp. 361–376.

[195] S. Sankar, Y. Zhang, S. Gurumurthi, and M. Stan, "Sensitivity-based optimization of disk architecture," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 69–81, Jan. 2009.

[196] I. Sato *et al.*, "Characteristics of heat transfer in small disk enclosures at high rotation speeds," *IEEE Trans. Compon., Hybrids, Manuf. Technol.*, vol. 13, no. 4, pp. 1006–1011, Dec. 1990.

[197] A. Hylick, R. Sohan, A. Rice, and B. Jones, "An analysis of hard drive energy consumption," in *Proc. IEEE Int. Symp. MASCOTS*, Sep. 2008, pp. 1–10.

[198] A. Hylick and R. Sohan, "A methodology for generating disk drive energy models using performance data," *Energy (Joules)*, vol. 80, p. 100, 2009.

[199] J. Zedlewski *et al.*, "Modeling hard-disk power consumption," in *Proc. 2nd USENIX Conf. FAST*, 2003, pp. 217–230.

[200] Q. Zhu *et al.*, "Hibernator: Helping disk arrays sleep through the winter," *SIGOPS Oper. Syst. Rev.*, vol. 39, no. 5, pp. 177–190, Oct. 2005.

[201] T. Bostoen, S. Mullender, and Y. Berbers, "Analysis of disk power management for data-center storage systems," in *Proc. 3rd Int. Conf. Future e-Energy Syst.—Where Energy, Comput. Commun. Meet*, May 2012, pp. 1–10.

[202] Y. Deng, "What is the future of disk drives, death or rebirth?" *ACM Comput. Surveys*, vol. 43, no. 3, pp. 23:1–23:27, Apr. 2011.

[203] K. Kant, "Data center evolution: A tutorial on state of the art, issues, and challenges," *Comput. Netw.*, vol. 53, no. 17, pp. 2939–2965, Dec. 2009.

[204] D. Andersen and S. Swanson, "Rethinking flash in the data center," *IEEE Micro*, vol. 30, no. 4, pp. 52–54, Jul. 2010.

[205] J. Park, S. Yoo, S. Lee, and C. Park, "Power modeling of solid state disk for dynamic power management policy design in embedded systems," in *Software Technologies for Embedded and Ubiquitous Systems*,

ser. Lecture Notes in Computer Science, vol. 5860, S. Lee and P. Narasimhan, Eds. Berlin, Germany: Springer-Verlag, 2009, pp. 24–35.

[206] V. Mohan *et al.*, "Modeling power consumption of nand flash memories using flashpower," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 7, pp. 1031–1044, Jul. 2013.

[207] S. J. E. Wilton and N. Jouppi, "CACTI: An enhanced cache access and cycle time model," *IEEE J. Solid-State Circuits*, vol. 31, no. 5, pp. 677–688, May 1996.

[208] Z. Li, K. M. Greenan, A. W. Leung, and E. Zadok, "Power consumption in enterprise-scale backup storage systems," in *Proc. 10th USENIX Conf. FAST*, 2012, pp. 6–6.

[209] M. Allalouf *et al.*, "Storage modeling for power estimation," in *Proc. SYSTOR—Israeli Exp. Syst. Conf.*, 2009, pp. 3:1–3:10.

[210] T. Inoue, A. Aikebaier, T. Enokido, and M. Takizawa, "A power consumption model of a storage server," in *Proc. 14th Int. Conf. NBiS*, Sep. 2011, pp. 382–387.

[211] A. Gandhi, M. Harchol-Balter, and I. Adan, "Server farms with setup costs," *Perform. Eval.*, vol. 67, no. 11, pp. 1123–1138, Nov. 2010.

[212] M. Mazzucco, D. Dyachuk, and M. Dikaiakos, "Profit-aware server allocation for green internet services," in *Proc. IEEE Int. Symp. MASCOTS*, 2010, pp. 277–284.

[213] M. Mazzucco and D. Dyachuk, "Balancing electricity bill and performance in server farms with setup costs," *Future Gener. Comput. Syst.*, vol. 28, no. 2, pp. 415–426, Feb. 2012.

[214] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *Proc. 11th SIGMETRICS Int. Joint Conf. Meas. Model. Comput. Syst.*, 2009, pp. 157–168.

[215] R. Lent, "Analysis of an energy proportional data center," *Ad Hoc Netw.*, vol. 25, Part B, pp. 554–564, Feb. 2015.

[216] I. Mitrani, "Trading power consumption against performance by reserving blocks of servers," in *Computer Performance Engineering*, vol. 7587, ser. Lecture Notes in Computer Science, M. Tribastone and S. Gilmore, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 1–15.

[217] Z. Liu *et al.*, "Renewable and cooling aware workload management for sustainable data centers," in *Proc. 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Meas. Model. Comput. Syst.*, 2012, pp. 175–186.

[218] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, Aug. 2009.

[219] M. Pedram, "Energy-efficient datacenters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 10, pp. 1465–1484, Oct. 2012.

[220] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling 'cool': Temperature-aware workload placement in data centers," in *Proc. Annu. Conf. USENIX ATEC*, 2005, p. 5.

[221] D. Duolikun, T. Enokido, A. Aikebaier, and M. Takizawa, "Energy-efficient dynamic clusters of servers," *J. Supercomput.*, vol. 71, no. 5, pp. 1–15, May 2014.

[222] A. Aikebaier, Y. Yang, T. Enokido, and M. Takizawa, "Energy-efficient computation models for distributed systems," in *Proc. Int. Conf. NBIS*, Aug. 2009, pp. 424–431.

[223] A. Aikebaier, T. Enokido, and M. Takizawa, "Distributed cluster architecture for increasing energy efficiency in cluster systems," in *Proc. ICPPW*, Sep. 2009, pp. 470–477.

[224] T. Inoue, A. Aikebaier, T. Enokido, and M. Takizawa, "Evaluation of an energy-aware selection algorithm for computation and storage-based applications," in *Proc. 15th Int. Conf. NBiS*, Sep. 2012, pp. 120–127.

[225] L. Velasco *et al.*, "Elastic operations in federated datacenters for performance and cost optimization," *Comput. Commun.*, vol. 50, pp. 142–151, Sep. 2014.

[226] D. A. Patterson, "Technical perspective: The data center is the computer," *Commun. ACM*, vol. 51, no. 1, pp. 105–105, Jan. 2008.

[227] B. Heller *et al.*, "Elastictree: Saving energy in data center networks," in *Proc. 7th USENIX Conf. NSDI*, 2010, pp. 17–17.

[228] C. Kachris and I. Tomkos, "Power consumption evaluation of all-optical data center networks," *Cluster Comput.*, vol. 16, no. 3, pp. 611–623, Sep. 2013.

[229] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 338–347, Jun. 2010.

[230] J. Shuja *et al.*, "Energy-efficient data centers," *Computing*, vol. 94, no. 12, pp. 973–994, Dec. 2012.

[231] Q. Yi and S. Singh, "Minimizing energy consumption of fattree data center networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 67–72, Dec. 2014.

[232] I. Widjaja, A. Walid, Y. Luo, Y. Xu, and H. Chao, "Small versus large: Switch sizing in topology design of energy-efficient data centers," in *Proc. IEEE/ACM 21st Int. Symp. IWQoS*, Jun. 2013, pp. 1–6.

[233] R. Tucker, "Green optical communications—Part II: Energy limitations in networks," *IEEE J. Sel. Topics Quantum Electron.*, vol. 17, no. 2, pp. 261–274, Mar. 2011.

[234] K. Hinton, G. Raskutti, P. Farrell, and R. Tucker, "Switching energy and device size limits on digital photonic signal processing technologies," *IEEE J. Sel. Topics Quantum Electron.*, vol. 14, no. 3, pp. 938–945, May 2008.

[235] Y. Zhang and N. Ansari, "Hero: Hierarchical energy optimization for data center networks," *IEEE Syst. J.*, vol. 9, no. 2, pp. 406–415, Jun. 2015.

[236] H. Jin *et al.*, "Joint host-network optimization for energy-efficient data center networking," in *Proc. IEEE 27th Int. Symp. IPDPS*, May 2013, pp. 623–634.

[237] D. Li, Y. Shang, and C. Chen, "Software defined green data center network with exclusive routing," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1743–1751.

[238] L. Niccolini, G. Iannaccone, S. Ratnasamy, J. Chandrashekar, and L. Rizzo, "Building a power-proportional software router," in *Proc. USENIX Conf. ATC*, 2012, p. 8.

[239] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *Proc. 8th Int. IFIP-TC 6 NETWORKING Conf.*, 2009, pp. 795–808.

[240] Cisco, "Cisco nexus 9500 platform switches," Cisco Nexus 9000 Series Switches, 2014.

[241] E. Bonetto, A. Finamore, M. Mellia, and R. Fiandra, "Energy efficiency in access and aggregation networks: From current traffic to potential savings," *Comput. Netw.*, vol. 65, pp. 151–166, Jun. 2014.

[242] A. Vishwanath, K. Hinton, R. Ayre, and R. Tucker, "Modelling energy consumption in high-capacity routers and switches," *IEEE. J. Sel. Areas Commun.*, vol. 32, no. 8, pp. 1524–1532, Aug. 2014.

[243] D. C. Kilper and R. S. Tucker, "Chapter 17—Energy-efficient telecommunications," in *Optical Fiber Telecommunications (Sixth Edition)*, 6th ed., ser. Optics and Photonics, I. P. Kaminow, T. Li, and A. E. Willner, Eds. Boston, MA, USA: Academic, 2013, pp. 747–791.

[244] F. Jalali *et al.*, "Energy consumption of photo sharing in online social networks," in *Proc. IEEE/ACM 14th Int. Symp. CCGrid*, May 2014, pp. 604–611.

[245] H. Hlavacs, G. Da Costa, and J. Pierson, "Energy consumption of residential and professional switches," in *Proc. Int. Conf. CSE*, Aug. 2009, vol. 1, pp. 240–246.

[246] P. Mahadevan, S. Banerjee, and P. Sharma, "Energy proportionality of an enterprise network," in *Proc. 1st ACM SIGCOMM Workshop Green Netw.*, 2010, pp. 53–60.

[247] J. Ahn and H.-S. Park, "Measurement and modeling the power consumption of router interface," in *Proc. 16th ICACT*, Feb. 2014, pp. 860–863.

[248] R. Bolla *et al.*, "The green abstraction layer: A standard power-management interface for next-generation network devices," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 82–86, Mar. 2013.

[249] R. Bolla, R. Bruschi, O. M. J. Ortiz, and P. Lago, "The energy consumption of TCP," in *Proc. 4th Int. Conf. Future e-Energy Syst.*, 2013, pp. 203–212.

[250] R. Basmadjian, H. Meer, R. Lent, and G. Giuliani, "Cloud computing and its interest in saving energy: The use case of a private cloud," *J. Cloud Comput.*, vol. 1, no. 1, p. 5, Jun. 2012.

[251] C. Kachris and I. Tomkos, "Power consumption evaluation of hybrid WDM PON networks for data centers," in *Proc. 16th Eur. Conf. NOC*, Jul. 2011, pp. 118–121.

[252] W. Van Heddeghem *et al.*, "Power consumption modeling in optical multilayer networks," *Photon. Netw. Commun.*, vol. 24, no. 2, pp. 86–102, Oct. 2012.

[253] M. McGarry, M. Reisslein, and M. Maier, "WDM ethernet passive optical networks," *IEEE Commun. Mag.*, vol. 44, no. 2, pp. 15–22, Feb. 2006.

[254] T. Shimada, N. Sakurai, and K. Kumozaki, "WDM access system based on shared demultiplexer and mmf links," *J. Lightw. Technol.*, vol. 23, no. 9, pp. 2621–2628, Sep. 2005.

[255] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch, and J. Underwood, "Power routing: Dynamic power provisioning in the data center," *SIGPLAN Notices*, vol. 45, no. 3, pp. 231–242, Mar. 2010.

[256] E. Oró, V. Depoorter, A. Garcia, and J. Salom, "Energy efficiency and renewable energy integration in data centres. strategies and modelling review," *Renewable Sustainable Energy Rev.*, vol. 42, pp. 429–445, Feb. 2015.

[257] D. Bouley and W. Torell, *Containerized Power and Cooling Modules for Data Centers*. West Kingston, RI, USA: American Power Conversion, 2012.

[258] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini, "Statistical profiling-based techniques for effective power provisioning in data centers," in *Proc. 4th ACM EuroSys Conf. Comput.*, 2009, pp. 317–330.

[259] H. Luo, B. Khargharia, S. Hariri, and Y. Al-Nashif, "Autonomic green computing in large-scale data centers," in *Energy-Efficient Distributed Computing Systems*. Hoboken, NJ, USA: Wiley, 2012, pp. 271–299.

[260] X. Fu, X. Wang, and C. Lefurgy, "How much power oversubscription is safe and allowed in data centers," in *Proc. 8th ACM ICAC*, 2011, pp. 21–30.

[261] D. Wang, C. Ren, and A. Sivasubramaniam, "Virtualizing power distribution in datacenters," in *Proc. 40th Annu. ISCA*, 2013, pp. 595–606.

[262] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, "Understanding and abstracting total data center power," in *Proc. Workshop Energy-Efficient Des.*, 2009, pp. 1–6.

[263] N. Rasmussen, *Electrical Efficiency Modeling of Data Centers*. West Kingston, RI, USA: American Power Conversion, 2006.

[264] P. Anderson, G. Backhouse, D. Curtis, S. Redding, and D. Wallom, *Low Carbon Computing: A View to 2050 and Beyond*. Bristol, U.K.: JISC, 2009.

[265] E. Lee, I. Kulkarni, D. Pompili, and M. Parashar, "Proactive thermal management in green datacenters," *J. Supercomput.*, vol. 60, no. 2, pp. 165–195, May 2012.

[266] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, "TACOMA: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 2, pp. 11:1–11:37, Jun. 2012.

[267] N. Rasmussen, *Calculating Total Cooling Requirements for Data Centers*, vol. 25, White Paper. West Kingston, RI, USA: American Power Conversion, 2007, pp. 1–8.

[268] E. Pakbaznia and M. Pedram, "Minimizing data center cooling and server power costs," in *Proc. 14th ACM/IEEE ISLPED*, 2009, pp. 145–150.

[269] A. Vasan, A. Sivasubramaniam, V. Shimpi, T. Sivabalan, and R. Subbiah, "Worth their watts?—An empirical study of datacenter servers," in *Proc. IEEE 16th Int. Symp. HPCA*, Jan. 2010, pp. 1–10.

[270] C. Lefurgy *et al.*, "Energy management for commercial servers," *Computer*, vol. 36, no. 12, pp. 39–48, Dec. 2003.

[271] J. Kim, M. Sabry, D. Atienza, K. Vaidyanathan, and K. Gross, "Global fan speed control considering non-ideal temperature measurements in enterprise servers," in *Proc. DATE*, Mar. 2014, pp. 1–6.

[272] M. Zapater *et al.*, "Leakage and temperature aware server control for improving energy efficiency in data centers," in *Proc. DATE*, Mar. 2013, pp. 266–269.

[273] D. Meisner and T. F. Wenisch, "Does low-power design imply energy efficiency for data centers?" in *Proc. 17th IEEE/ACM ISLPED*, 2011, pp. 109–114.

[274] O. Mämmelä *et al.*, "Energy-aware job scheduler for high-performance computing," *Comput. Sci. Res. Develop.*, vol. 27, no. 4, pp. 265–275, Nov. 2012.

[275] Cisco Energy Efficient Data Center Solutions and Best Practices, Cisco, San Jose, CA, USA, 2007.

[276] M. Patterson, "The effect of data center temperature on energy efficiency," in *Proc. 11th Intersoc. Conf. ITHERM Phenom. Electron. Syst.*, May 2008, pp. 1167–1174.

[277] R. Ghosh, V. Sundaralingam, and Y. Joshi, "Effect of rack server population on temperatures in data centers," in *Proc. IEEE 13th Intersoc. Conf. ITherm Phenom. Electron. Syst.*, May 2012, pp. 30–37.

[278] J. Dai, M. Ohadi, D. Das, and M. Pecht, "The telecom industry and data centers," in *Optimum Cooling of Data Centers*. New York, NY, USA: Springer-Verlag, 2014, pp. 1–8.

[279] H. Ma and C. Chen, "Development of a divided zone method for power savings in a data center," in *Proc. IEEE 29th Annu. SEMI-THERM*, Mar. 2013, pp. 33–38.

[280] M. David and R. Schmidt, "Impact of ASHRAE environmental classes on data centers," in *Proc. IEEE Intersoc. Conf. ITherm Phenom. Electron. Syst.*, May 2014, pp. 1092–1099.

[281] S. K. S. Gupta *et al.*, "GDCSIM: A simulator for green data center design and analysis," *ACM Trans. Model. Comput. Simul.*, vol. 24, no. 1, pp. 3:1–3:27, Jan. 2014.

[282] T. Malkamaki and S. Ovaska, "Data centers and energy balance in Finland," in *Proc. IGCC*, Jun. 2012, pp. 1–6.

[283] X. Zhan and S. Reda, "Techniques for energy-efficient power budgeting in data centers," in *Proc. 50th Annu. DAC*, 2013, pp. 176:1–176:7.

[284] J. Doyle, R. Shorten, and D. O'Mahony, "Stratus: Load balancing the cloud for carbon emissions control," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, p. 1, Jan. 2013.

[285] R. Das, J. O. Kephart, J. Lenchner, and H. Hamann, "Utility-function-driven energy-efficient cooling in data centers," in *Proc. 7th ICAC*, 2010, pp. 61–70.

[286] R. Das, S. Yarlanki, H. Hamann, J. O. Kephart, and V. Lopez, "A unified approach to coordinated energy-management in data centers," in *Proc. 7th Int. CNSM*, 2011, pp. 504–508.

[287] H. Hamann, M. Schappert, M. Iyengar, T. van Kessel, and A. Claassen, "Methods and techniques for measuring and improving data center best practices," in *Proc. 11th ITHERM Phenom. Electron. Syst.*, May 2008, pp. 1146–1152.

[288] H. Hamann *et al.*, "Uncovering energy-efficiency opportunities in data centers," *IBM J. Res. Develop.*, vol. 53, no. 3, pp. 10:1–10:12, May 2009.

[289] R. T. Kaushik and K. Nahrstedt, "T: A data-centric cooling energy costs reduction approach for big data analytics cloud," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. SC*, 2012, pp. 52:1–52:11.

[290] X. Han and Y. Joshi, "Energy reduction in server cooling via real time thermal control," in *Proc. IEEE 28th Annu. SEMI-THERM Meas. Manage. Symp.*, Mar. 2012, pp. 20–27.

[291] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *Proc. 4th Int. Conf. Future e-Energy Syst.*, 2013, pp. 137–148.

[292] X. Zheng and Y. Cai, "Energy-aware load dispatching in geographically located internet data centers," *Sustain. Comput., Informat. Syst.*, vol. 1, no. 4, pp. 275–285, Dec. 2011.

[293] B. Whitehead, D. Andrews, A. Shah, and G. Maidment, "Assessing the environmental impact of data centres—Part 2: Building environmental assessment methods and life cycle assessment," *Building Environ.*, vol. 93, pp. 395–405, Nov. 2015.

[294] L. Wang and S. Khan, "Review of performance metrics for green data centers: A taxonomy study," *J. Supercomput.*, vol. 63, no. 3, pp. 639–656, Mar. 2013.

[295] P. Rad, M. Thoene, and T. Webb, "Best practices for increasing data center energy efficiency," *Dell Power Sol. Mag.*, pp. 1–5, Feb. 2008.

[296] Pue: A Comprehensive Examination of the Metric, Green Grid, Beaverton, OR, USA, 2012.

[297] A. Khosravi, S. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Euro-Par 2013 Parallel Processing*, vol. 8097, ser. Lecture Notes in Computer Science, F. Wolf, B. Mohr, and D. Mey, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 317–328.

[298] J. Gao, "Machine learning applications for data center optimization," Google White Paper, 2014.

[299] J. Dai, M. Ohadi, D. Das, and M. Pecht, "Data center energy flow and efficiency," in *Optimum Cooling of Data Centers*. New York, NY, USA: Springer-Verlag, 2014, pp. 9–30.

[300] K. Choo, R. M. Galante, and M. M. Ohadi, "Energy consumption analysis of a medium-size primary data center in an academic campus," *Energy Buildings*, vol. 76, pp. 414–421, Jun. 2014.

[301] R. Zhou, Y. Shi, and C. Zhu, "Axpue: Application level metrics for power usage effectiveness in data centers," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2013, pp. 110–117.

[302] R. Giordanelli, C. Mastroianni, M. Meo, G. Papuzzo, and A. Roscetti, *Saving Energy in Data Centers*. Rende, Italy: Eco4Cloud, 2013.

[303] G. A. Brady, N. Kapur, J. L. Summers, and H. M. Thompson, "A case study and critical assessment in calculating power usage effectiveness for a data centre," *Energy Convers. Manage.*, vol. 76, pp. 155–161, Dec. 2013.

[304] J. Yuventi and R. Mehdizadeh, "A critical analysis of power usage effectiveness and its use in communicating data center energy consumption," *Energy Buildings*, vol. 64, pp. 90–94, Sep. 2013.

[305] Green Grid Data Center Power Efficiency Metrics: Pue and DCIE, Green Grid, Beaverton, OR, USA, 2008.

[306] D. Chernicoff, *The Shortcut Guide to Data Center Energy Efficiency*. Realtime Publ., 2009.

[307] M. Patterson *et al.*, "Tue, a new energy-efficiency metric applied at Ornl's jaguar," in *Supercomputing*, vol. 7905, ser. Lecture Notes in Computer Science, J. Kunkel, T. Ludwig, and H. Meuer, Eds. Berlin-Verlag: Springer-Verlag, 2013, pp. 372–382.

[308] Introduction of Datacenter Performance per Energy, Green IT Promotion Council, Tokyo, Japan, 2010.

[309] New Data Center Energy Efficiency Evaluation Index Dppe(Datacenter Performance per Energy) Measurement Guidelines (ver 2.05), Green IT Promotion Council, Tokyo, Japan, 2012.

[310] M. Obaidat, A. Anpalagan, and I. Woungang, *Handbook of Green Information and Communication Systems*. Amsterdam, The Netherlands: Elsevier, 2012.

[311] L. H. Sego *et al.*, "Implementing the data center energy productivity metric," *J. Emerging Technol. Comput. Syst.*, vol. 8, no. 4, pp. 30:1–30:22, Nov. 2012.

[312] P. Mathew, S. Greenberg, D. Sartor, J. Bruschi, and L. Chu, *Self-Benchmarking Guide for Data Center Infrastructure: Metrics, Benchmarks, Actions*. Berkeley, CA, USA: Lawrence Berkeley National Laboratory, 2010.

[313] T. Wilde *et al.*, "DWPE, a new data center energy-efficiency metric bridging the gap between infrastructure and workload," in *Proc. Int. Conf. HPCS*, Jul. 2014, pp. 893–901.

[314] B. Aebischer and L. Hilty, "The energy demand of ICT: A historical perspective and current methodological challenges," in *ICT Innovations for Sustainability*, vol. 310, ser. Advances in Intelligent Systems and Computing. Berlin, Germany: Springer-Verlag, 2015, pp. 71–103.

[315] G. Le Louet and J.-M. Menaud, "Optiplace: Designing cloud management with flexible power models through constraint programing," in *Proc. IEEE/ACM 6th Int. Conf. UCC*, Dec. 2013, pp. 211–218.

[316] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No 'power' struggles: Coordinated multi-level power management for the data center," *SIGARCH Comput. Archit. News*, vol. 36, no. 1, pp. 48–59, Mar. 2008.

[317] M. Islam, S. Ren, and G. Quan, "Online energy budgeting for virtualized data centers," in *Proc. IEEE 21st Int. Symp. MASCOTS*, Aug. 2013, pp. 424–433.

[318] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *Proc. ACM SIGMETRICS Joint Int. Conf. Meas. Model. Comput. Syst.*, 2011, pp. 233–244.

[319] M. Lin, Z. Liu, A. Wierman, and L. Andrew, "Online algorithms for geographical load balancing," in *Proc. IGCC*, Jun. 2012, pp. 1–10.

[320] E. Masanet, R. Brown, A. Shehabi, J. Koomey, and B. Nordman, "Estimating the energy use and efficiency potential of U.S. data centers," *Proc. IEEE*, vol. 99, no. 8, pp. 1440–1453, Aug. 2011.

[321] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Power cost reduction in distributed data centers: A two-time-scale approach for delay tolerant workloads," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 200–211, Jan. 2014.

[322] A. H. Mahmud and S. Ren, "Online capacity provisioning for carbon-neutral data center with demand-responsive electricity prices," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 2, pp. 26–37, Aug. 2013.

[323] Z. Zhou *et al.*, "Carbon-aware load balancing for geo-distributed cloud services," in *Proc. IEEE 21st Int. Symp. MASCOTS*, Aug. 2013, pp. 232–241.

[324] Z. Liu, I. Liu, S. Low, and A. Wierman, "Pricing data center demand response," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2014, pp. 111–123.

[325] J. Davis, S. Rivoire, M. Goldszmidt, and E. Ardestani, "Chaos: Composable highly accurate OS-based power models," in *Proc. IEEE IISWC*, Nov. 2012, pp. 153–163.

[326] J. Davis, S. Rivoire, and M. Goldszmidt, "Star-Cap: Cluster power management using software-only models," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2012-107.

[327] O. Laadan and J. Nieh, "Transparent checkpoint-restart of multiple processes on commodity operating systems," in *Proc. USENIX ATC*, 2007, pp. 25:1–25:14.

[328] J. Stoess, C. Lang, and F. Bellosa, "Energy management for hypervisor-based virtual machines," in *Proc. USENIX ATC*, 2007, pp. 1:1–1:14.

[329] N. Kim, J. Cho, and E. Seo, "Energy-based accounting and scheduling of virtual machines in a cloud system," in *Proc. IEEE/ACM Int. Conf. GreenCom*, Aug. 2011, pp. 176–181.

[330] F. Quesnel, H. Mehta, and J.-M. Menaud, "Estimating the power consumption of an idle virtual machine," in *Proc. IEEE GreenCom/iThings/CPSCom/IEEE Cyber, Phys. Social Comput.*, Aug. 2013, pp. 268–275.

[331] F. Chen, J. Grundy, J.-G. Schneider, Y. Yang, and Q. He, "Automated analysis of performance and energy consumption for cloud applications," in *Proc. 5th ACM/SPEC ICPE*, 2014, pp. 39–50.

[332] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch, "Power management of online data-intensive services," in *Proc. 38th Annu. ISCA*, 2011, pp. 319–330.

[333] M. Poess and R. Othayoth Nambiar, "A power consumption analysis of decision support systems," in *Proc. 1st Joint WOSP/SIPEW*, 2010, pp. 147–152.

[334] M. Horiuchi and K. Taura, "Acceleration of data-intensive workflow applications by using file access history," in *Proc. SCC High Perform. Comput., Netw., Storage Anal.*, Nov. 2012, pp. 157–165.

[335] M. Gamell, I. Rodero, M. Parashar, and S. Poole, "Exploring energy and performance behaviors of data-intensive scientific workflows on systems with deep memory hierarchies," in *Proc. 20th Int. Conf. HiPC*, Dec. 2013, pp. 226–235.

[336] L. Mashayekhy, M. Nejad, D. Grosu, D. Lu, and W. Shi, "Energy-aware scheduling of MapReduce jobs," in *Proc. IEEE Int. Conf. BigData Congr.*, Jun. 2014, pp. 32–39.

[337] L. Mashayekhy, M. Nejad, D. Grosu, Q. Zhang, and W. Shi, "Energy-aware scheduling of MapReduce jobs for big data applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 10, pp. 2720–2733, Oct. 2014.

[338] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[339] T. White, *Hadoop: The Definitive Guide*. Newton, MA, USA: O'Reilly Media, Inc., 2012.

[340] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: Distributed data-parallel programs from sequential building blocks," in *Proc. 2nd ACM SIGOPS/EuroSys Conf. Comput.*, 2007, pp. 59–72.

[341] N. Zhu, L. Rao, X. Liu, J. Liu, and H. Guan, "Taming power peaks in MapReduce clusters," in *Proc. ACM SIGCOMM Conf.*, 2011, pp. 416–417.

[342] N. Zhu, L. Rao, X. Liu, and J. Liu, "Handling more data with less cost: Taming power peaks in MapReduce clusters," in *Proc. APSYS Workshop*, 2012, pp. 3:1–3:6.

[343] N. Zhu, X. Liu, J. Liu, and Y. Hua, "Towards a cost-efficient MapReduce: Mitigating power peaks for hadoop clusters," *Tsinghua Sci. Technol.*, vol. 19, no. 1, pp. 24–32, Feb. 2014.

[344] B. Feng, J. Lu, Y. Zhou, and N. Yang, "Energy efficiency for MapReduce workloads: An in-depth study," in *Proc. 23rd ADC*, Darlinghurst, Australia, 2012, vol. 124, pp. 61–70.

[345] W. Lang and J. M. Patel, "Energy management for MapReduce clusters," *Proc. VLDB Endowment*, vol. 3, no. 1/2, pp. 129–139, Sep. 2010.

[346] A. Di Stefano, G. Morana, and D. Zito, "Improving the allocation of communication-intensive applications in clouds using time-related information," in *Proc. 11th ISPDC*, Jun. 2012, pp. 71–78.

[347] P. Pacheco, *An Introduction to Parallel Programming*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[348] M. E. M. Diouri, O. Glück, J.-C. Mignot, and L. Lefèvre, "Energy estimation for mpi broadcasting algorithms in large scale HPC systems," in *Proc. 20th EuroMPI Users' Group Meet.*, 2013, pp. 111–116.

[349] M. Gamell *et al.*, "Exploring power behaviors and trade-offs of in-situ data analytics," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. SC*, 2013, pp. 77:1–77:12.

[350] C. Bunse and S. Stiemer, "On the energy consumption of design patterns," *Softwaretechnik-Trends*, vol. 33, no. 2, pp. 1–2, 2013.

[351] J. Arjona Aroca, A. Chatzipapas, A. Fernández Anta, and V. Mancuso, "A measurement-based analysis of the energy consumption of data center servers," in *Proc. 5th Int. Conf. Future e-Energy Syst.*, 2014, pp. 63–74.

[352] S. Wang, Y. Li, W. Shi, L. Fan, and A. Agrawal, "Safari: Function-level power analysis using automatic instrumentation," in *Proc. Int. Conf. Energy Aware Comput.*, Dec. 2012, pp. 1–6.

[353] CoolEmAll Project, Coolemall, 2014. [Online]. Available: http://www.coolemall.eu

[354] L. Cupertino *et al.*, "Energy-efficient, thermal-aware modeling and simulation of data centers: The coolemall approach and evaluation results," *Ad Hoc Netw.*, vol. 25, Part B, pp. 535–553, Feb. 2015.

[355] C. Seo, G. Edwards, D. Popescu, S. Malek, and N. Medvidovic, "A framework for estimating the energy consumption induced by a distributed system's architectural style," in *Proc. 8th Int. Workshop SAVCBS*, 2009, pp. 27–34.

[356] C. Seo, S. Malek, and N. Medvidovic, "An energy consumption framework for distributed java-based systems," in *Proc. 20nd IEEE/ACM Int. Conf. ASE*, 2007, pp. 421–424.

[357] B. Cumming *et al.*, "Application centric energy-efficiency study of distributed multi-core and hybrid CPU-GPU systems," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. SC*, 2014, pp. 819–829.

[358] R. Koller, A. Verma, and A. Neogi, "Wattapp: An application aware power meter for shared data centers," in *Proc. 7th ICAC*, 2010, pp. 31–40.

[359] J. Demmel, A. Gearhart, B. Lipshitz, and O. Schwartz, "Perfect strong scaling using no additional energy," in *Proc. IEEE 27th Int. Symp. IPDPS*, May 2013, pp. 649–660.

[360] P. Bartalos and M. Blake, "Engineering energy-aware web services toward dynamically-green computing," in *Service-Oriented Computing—ICSOC 2011 Workshops*, vol. 7221, ser. Lecture Notes in Computer Science, G. Pallis, M. Jmaiel, A. Charfi, S. Graupner, Y. Karabulut, S. Guinea, F. Rosenberg, Q. Sheng, C. Pautasso, and S. Mokhtar, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 87–96.

[361] P. Bartalos and M. Blake, "Green web services: Modeling and estimating power consumption of web services," in *Proc. IEEE 19th ICWS*, 2012, pp. 178–185.

[362] A. Nowak, T. Binz, F. Leymann, and N. Urbach, "Determining power consumption of business processes and their activities to enable green business process reengineering," in *Proc. IEEE 17th Int. EDOC Conf.*, Sep. 2013, pp. 259–266.

[363] J. Berral, R. Gavalda, and J. Torres, "Power-aware multi-data center management using machine learning," in *Proc. 42nd ICPP*, Oct. 2013, pp. 858–867.

[364] S. Marsland, *Machine Learning: An Algorithmic Perspective*. New York, NY, USA: Taylor & Francis, 2011.

[365] K. Cios, R. Swiniarski, W. Pedrycz, and L. Kurgan, "Supervised learning: Decision trees, rule algorithms, and their hybrids," in *Data Mining*. New York, NY, USA: Springer-Verlag, 2007, pp. 381–417.

[366] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[367] J. Berral, R. Gavalda, and J. Torres, "Adaptive scheduling on power-aware managed data-centers using machine learning," in *Proc. IEEE/ACM 12th Int. Conf. GRID Comput.*, Sep. 2011, pp. 66–73.

[368] J. Dolado, D. Rodríguez, J. Riquelme, F. Ferrer-Troyano, and J. Cuadrado, "A two-stage zone regression method for global characterization of a project database," in *Advances in Machine Learning Applications in Software Engineering*. Hershey, PA, USA: IGI Global, 2007, p. 1.

[369] J. L. Berral *et al.*, "Towards energy-aware scheduling in data centers using machine learning," in *Proc. 1st Int. Conf. e-Energy-Efficient Comput. Netw.*, 2010, pp. 215–224.

[370] J. Ll. Berral, R. Gavaldà, and J. Torres, "Empowering automatic data-center management with machine learning," in *Proc. 28th Annu. ACM SAC*, 2013, pp. 170–172.

[371] J. L. Berral *et al.*, *Toward Energy-Aware Scheduling Using Machine Learning*. Hoboken, NJ, USA: Wiley, 2012, pp. 215–244.

[372] B. Khargharia, H. Luo, Y. Al-Nashif, and S. Hariri, "Appflow: Autonomic performance-per-watt management of large-scale data centers," in *Proc. IEEE/ACM Int. Conf. GreenCom/CPSCom*, Dec. 2010, pp. 103–111.

[373] H. Shen, Y. Tan, J. Lu, Q. Wu, and Q. Qiu, "Achieving autonomous power management using reinforcement learning," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 18, no. 2, pp. 24:1–24:32, Apr. 2013.

[374] F. Caglar and A. Gokhale, "iOverbook: Intelligent resource-overbooking to support soft real-time applications in the cloud," in *Proc. IEEE Int. Conf. CLOUD Comput.*, Jul. 2014, pp. 538–545.

[375] M. Guzek, S. Varrette, V. Plugaru, J. E. Pecero, and P. Bouvry, "A holistic model of the performance and the energy efficiency of hypervisors in a high-performance computing environment," *Concurr. Comput., Pract. Exp.*, vol. 26, no. 15, pp. 2569–2590, Oct. 2014.

[376] G. Tesauro *et al.*, "Managing power consumption and performance of computing systems using reinforcement learning," in *Proc. NIPS*, 2007, pp. 1–8.

[377] A. A. Bhattacharya, D. Culler, A. Kansal, S. Govindan, and S. Sankar, "The need for speed and stability in data center power capping," in *Proc. IGCC*, 2012, pp. 1–10.

[378] Q. Li *et al.*, "An embedded software power model based on algorithm complexity using back-propagation neural networks," in *Proc. IEEE/ACM GreenCom/CPSCom*, Dec. 2010, pp. 454–459.

[379] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 155–162, Jan. 2012.

[380] Y. Gao *et al.*, "Service level agreement based energy-efficient resource management in cloud data centers," *Comput. Elect. Eng.*, vol. 40, no. 5, pp. 1621–1633, Jul. 2013.

[381] Z. Zhang and S. Fu, "Macropower: A coarse-grain power profiling framework for energy-efficient cloud computing," in *Proc. IEEE 30th IPCCC*, 2011, pp. 1–8.

[382] C. Bunse, H. Höpfner, S. Klingert, E. Mansour, and S. Roychoudhury, "Energy aware database management," in *Energy-Efficient Data Centers*, vol. 8343, ser. Lecture Notes in Computer Science, S. Klingert, X. Hesselbach-Serra, M. Ortega, and G. Giuliani, Eds. Berlin Heidelberg: Springer-Verlag, 2014, pp. 40–53.

[383] H. Höpfner and C. Bunse, "Energy aware data management on avr micro controller based systems," *SIGSOFT Softw. Eng. Notes*, vol. 35, no. 3, pp. 1–8, May 2010.

[384] FPGA Power Management and Modeling Techniques, Altera, San Jose, CA, USA, 2012.

[385] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics of field programmable gate arrays," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 11, pp. 1712–1724, Nov. 2005.

[386] K. K. W. Poon, S. J. E. Wilton, and A. Yan, "A detailed power model for field-programmable gate arrays," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 10, no. 2, pp. 279–302, Apr. 2005.

[387] S. Dawson-Haggerty, A. Krioukov, and D. E. Culler, "Power optimization—A reality check," EECS Department, Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-140, Oct. 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-140.html

[388] Y. Zhang and N. Ansari, "On architecture design, congestion notification, tcp incast and power consumption in data centers," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 39–64, 1st Quart. 2013.

[389] V. Tiwari, S. Malik, A. Wolfe, and M.-C. Lee, "Instruction level power analysis and optimization of software," in *Proc. 9th Int. Conf. VLSI Des.*, Jan. 1996, pp. 326–328.

[390] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad, "Exploring smart grid and data center interactions for electric power load balancing," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 3, pp. 89–94, Jan. 2014.

[391] D. Barbagallo, E. Di Nitto, D. J. Dubois, and R. Mirandola, "A bio-inspired algorithm for energy optimization in a self-organizing data center," in *Proc. 1st Int. Conf. SOAR*, 2010, pp. 127–151.

[392] R. Carroll *et al.*, "Bio-inspired service management framework: Green data-centres case study," *Int. J. Grid Utility Comput.*, vol. 4, no. 4, pp. 278–292, Oct. 2013.

[393] D. J. Dubois, "Self-organizing methods and models for software development," Ph.D. dissertation, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, 2010.

[394] L. Gyarmati and T. Trinh, "Energy efficiency of data centers," in *Green IT: Technologies and Applications*, J. Kim and M. Lee, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 229–244.

[395] S. Dehuri, S. Ghosh, and S. Cho, *Integration of Swarm Intelligence and Artificial Neutral Network*, ser. Series in machine perception and artificial intelligence. Singapore: World Scientific, 2011.

[396] C. B. Pop, I. Anghel, T. Cioara, I. Salomie, and I. Vartic, "A swarm-inspired data center consolidation methodology," in *Proc. 2nd Int. Conf. WIMS*, 2012, pp. 41:1–41:7.

[397] I. Anghel, C. B. Pop, T. Cioara, I. Salomie, and I. Vartic, "A swarm-inspired technique for self-organizing and consolidating data centre servers," *Scalable Comput., Pract. Exp.*, vol. 14, no. 2, pp. 69–82, 2013.

[398] E. Feller, L. Rilling, and C. Morin, "Energy-aware ant colony based workload placement in clouds," in *Proc. IEEE/Acm 12th Int. Conf. GRID Comput.*, Sep. 2011, pp. 26–33.

[399] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[400] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[401] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project adam: Building an efficient and scalable deep learning training system," in *Proc 11th USENIX Symp. OSDI*, Oct. 2014, pp. 571–582.

[402] C. Li, R. Wang, T. Li, D. Qian, and J. Yuan, "Managing green datacenters powered by hybrid renewable energy systems," in *Proc. 11th ICAC*, 2014, pp. 261–272.

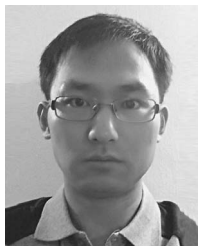
[403] I. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and greenswitch: Managing datacenters powered by renewable energy," in *Proc. 18th Int. Conf. ASPLOS*, 2013, pp. 51–64.

[404] R. Weidmann and H.-R. Vogel, "Data center 2.0: Energy-efficient and sustainable," in *The Road to a Modern IT Factory*, ser. Management for Professionals, F. Abolhassan, Ed. Berlin, Germany: Springer-Verlag, 2014, pp. 129–136.

[405] B. Aksanli, E. Pettis, and T. Rosing, "Architecting efficient peak power shaving using batteries in data centers," in *Proc. IEEE 21st Int. Symp. MASCOTS*, Aug. 2013, pp. 242–253.
[406] V. Kontorinis *et al.*, "Managing distributed UPS energy for effective power capping in data centers," *SIGARCH Comput. Archit. News*, vol. 40, no. 3, pp. 488–499, Jun. 2012.
[407] Y. Kuroda, A. Akai, T. Kato, and Y. Kudo, "High-efficiency power supply system for server machines in data center," in *Proc. Int. Conf. HPCS*, Jul. 2013, pp. 172–177.
[408] S. Hancock, Iceland Looks to Serve the World, 2009. [Online]. Available: http://news.bbc.co.uk/2/hi/programmes/click_online/8297237.stm
[409] W. Zheng, K. Ma, and X. Wang, "Exploiting thermal energy storage to reduce data center capital and operating expenses," in *Proc. IEEE 20th Int. Symp. HPCA*, Feb. 2014, pp. 132–141.
[410] L. Ganesh, H. Weatherspoon, T. Marian, and K. Birman, "Integrated approach to data center power management," *IEEE Trans. Comput.*, vol. 62, no. 6, pp. 1086–1096, Jun. 2013.

Miyuru Dayarathna received the B.Sc. (Hons.) degree in information technology from University of Moratuwa, Moratuwa, Sri Lanka, in 2008, the master's degree in media design from Keio University, Tokyo, Japan, in 2010, and the Ph.D. degree in computer science from Tokyo Institute of Technology, Japan, in 2013. He is a Research Fellow at the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. His research interests include stream computing, graph data management and mining, energy efficient computer systems, cloud computing, high performance computing, database systems, and performance engineering. He has published technical papers in various international journals and conferences.

Yonggang Wen (S'99–M'08–SM'14) received the Ph.D. degree in electrical engineering and computer science (minor in western literature) from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He is an Assistant Professor with the School of Computer Engineering at Nanyang Technological University, Singapore. Previously, he has worked in Cisco to lead product development in content delivery network, which had a revenue impact of $3 billon (U.S.) globally. He has published over 130 papers in top journals and prestigious conferences. His work in multi-screen cloud social TV has been featured by global media (more than 1600 news articles from over 29 countries) and received ASEAN ICT Award 2013 (Gold Medal). His work on Cloud3DView for Data Centre Life-Cycle Management, as the only academia entry, has made him one of the top 4 finalist in the Data Centre Dynamics Awards 2014—APAC. Dr. Wen is a co-recipient of Best Paper Awards at EAI Chinacom 2015, IEEE WCSP 2014, IEEE GLOBECOM 2013, and IEEE EUC 2012, and a co-recipient of 2015 IEEE Multimedia Best Paper Award. He serves on editorial boards for IEEE COMMUNICATIONS SURVEY & TUTORIALS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, IEEE ACCESS, and *Ad Hoc Networks* (Elsevier), and was elected as the Chair for IEEE ComSoc Multimedia Communication Technical Committee (2014–2016). His research interests include cloud computing, green data center, big data analytics, multimedia network, and mobile computing



Rui Fan received the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, and received MIT's Sprowls Award for best doctoral theses in computer science. He is an Assistant Professor at Nanyang Technological University, Singapore. His research interests include theoretical and applied algorithms for parallel and distributed computing, especially algorithms for multicores, GPUs, and cloud computing.