A Survey on Modeling Energy Consumption of Cloud Applications: Deconstruction, State of the Art, and Trade-Off Debates

Zheng Li, *Member, IEEE*, Selome Tesfatsion, *Student Member, IEEE*, Saeed Bastani, *Member, IEEE*, Ahmed Ali-Eldin, *Member, IEEE*, Erik Elmroth, *Member, IEEE*, Maria Kihl, *Member, IEEE*, and Rajiv Ranjan, *Member, IEEE*

Abstract—Given the complexity and heterogeneity in Cloud computing scenarios, the modeling approach has widely been employed to investigate and analyze the energy consumption of Cloud applications, by abstracting real-world objects and processes that are difficult to observe or understand directly. It is clear that the abstraction sacrifices, and usually does not need, the complete reflection of the reality to be modeled. Consequently, current energy consumption models vary in terms of purposes, assumptions, application characteristics and environmental conditions, with possible overlaps between different research works. Therefore, it would be necessary and valuable to reveal the state-of-the-art of the existing modeling efforts, so as to weave different models together to facilitate comprehending and further investigating application energy consumption in the Cloud domain. By systematically selecting, assessing, and synthesizing 76 relevant studies, we rationalized and organized over 30 energy consumption models with unified notations. To help investigate the existing models and facilitate future modeling work, we deconstructed the runtime execution and deployment environment of Cloud applications, and identified 18 environmental factors and 12 workload factors that would be influential on the energy consumption. In particular, there are complicated trade-offs and even debates when dealing with the combinational impacts of multiple factors.

Index Terms—Application energy consumption, cloud computing, energy consumption modeling, systematic literature review

1 Introduction

Given the requirement of efficient use of computing power and the increasing consideration of global warming, the energy consumption management is a crucial concern across the entire community of the information and communication technology (ICT), especially in the Cloud computing domain [1]. In particular, understanding Cloud applications' energy consumption has been identified to be a prerequisite for developing energy saving mechanisms [2]. Unfortunately, due to Cloud applications' inherent complexity and their environmental heterogeneity, it would be extremely challenging to tune the energy efficiency of a real-world application [3], and even unpractical to directly measure its energy consumption. On one hand, the components and data of a modern application could largely be distributed and spread in Cloud environments. On the other hand, the same computing resource in the Cloud could be shared among a bunch of different applications.

Consequently, most of the related work focused on the energy expense in the Cloud infrastructure and IT equipment (e.g., data center energy consumption [4], [5]), without considering specific application scenarios or isolating a single application from its surroundings. In particular, with a lack of concern about the application runtime, some of the studies essentially emphasized the power consumption in Cloud systems from the hardware's perspective (e.g., [6]). Note that here power (measured in *Watts*) is defined as the rate at which energy (measured in *Joules*) is consumed in the Cloud infrastructure.

To facilitate investigating Cloud applications' energy consumption, the modeling approach tends to be promising to relieve the aforementioned challenges and complexity, by abstracting real-world objects or processes that are difficult to observe or understand directly [7]. However, since such an abstraction sacrifices (and usually does not need) the complete reflection of the reality to be modeled, current energy consumption models vary in terms of purposes, assumptions, application characteristics and environmental conditions, with possible overlaps between different research works. As a result, different models need to be weaved together to reflect a full scope of energy consumption aspects, which is also common in other domains [8].

Therefore, to facilitate understanding the nature of the energy consumption of Cloud applications, it would be

- *Z. Li, S. Bastani, and M. Kihl are with the Department of Electrical and Information Technology, Lund University, Lund 223 63, Sweden. E-mail: {zheng.li, saeed.bastani, maria.kihl}@eit.lth.se.*
- *S. Tesfatsion, A. Ali-Eldin, and E. Elmroth are with the Department of Computing Science, Umeå University, Umeå 901 87, Sweden. E-mail: {selome, ahmeda, elmroth}@cs.umu.se.*
- *R. Ranjan is with the School of Computer Science, Newcastle University, Newcastle Upon Tyne NE1 7RU, United Kingdom. E-mail: raj.ranjan@ncl.ac.uk.*

Manuscript received 25 Feb. 2017; revised 18 June 2017; accepted 29 June 2017. Date of publication 3 July 2017; date of current version 6 Sept. 2017. (Corresponding author: Maria Kihl.)
Recommended for acceptance by K. Li.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSUSC.2017.2722822

useful and valuable to come up with the state-of-the-art of the existing modeling efforts that play an evidence role in revealing the reality. When it comes to the evidence aggregation for answering research questions in software engineering and computer science, a standard and rigorous methodology is Systematic Literature Review (SLR) [9]. Thus, we implemented an SLR to identify, examine and synthesize the existing models developed/employed in the relevant studies. Moreover, to help analyze and compare the existing models, we followed the divide-and-conquer strategy to also study the prerequisites of modeling practices: (1) Since the energy for running a Cloud application is driven by the combined mutual effects of the application and its environment [10], we extracted nine generic application execution elements and built up an evidence-based architecture of the application deployment environment. (2) Considering that Cloud computing scenarios involve numerous and various factors [11], we identified 18 environmental factors and 12 workload factors respectively as well as their individual influences on Cloud applications' energy consumption.

Driven by the aforementioned motivations, our main contributions to the research field can be summarized as follows. First, our deconstruction of Cloud application runtime and deployment environment offers an expandable dictionary of energy-related factors. Benefiting from this dictionary, researchers and practitioners can conveniently screen the existing concerns and choose suitable ones for new energy consumption studies. In fact, pre-listing all the domain-relevant factors has been considered to be a "tedious but crucial task" for factorial studies in general [12], [13]. Second, the systematically organized models with unified notations can act as a knowledge artefact for both researchers and practitioners to not only reveal the fundamentals of energy consumption, but also facilitate simulations to deal with a wide range of Cloud application energy efficiency problems. For example, accurate model-based energy consumption simulations would be significantly beneficial for decision making in various trade-off situations.

The remainder of this paper is organized as follows. Section 2 discusses the related work, and particularly explains our research scope. Section 3 briefly describes the methodology employed in our survey. Section 4 specifies the results of this survey by addressing the predefined research questions. Section 5 lists four trade-off debates to demonstrate both the complexity in combinational effects of multiple factors, and the potential research directions that can benefit from our survey. Conclusions and our future work are outlined in Section 6.

2 Related Work: A Brief Explanation of the Research Scope

The existing studies with respect to modeling the energy consumption of Cloud applications have various scenarios, purposes and contexts. Moreover, different aspects and stages (e.g., programming IDEs and Languages at design/implementation time [14], [15]) within the whole lifecycle of a Cloud application might all have impacts on its total energy consumption. However, exhaustively including all the energy-relevant application features could make this research out of control. Therefore, given the discussions about the dominant contributors [16], we are only concerned with the energy consumed for deploying and executing Cloud applications. To further shape the scope of this research, we narrow down the related work by applying a set of selection/exclusion criteria, and the keywords are "Cloud", "application", and "model".

First, we focus on the Cloud. As mentioned previously, the energy consumption has become a crucial concern across the entire ICT community. Therefore, there are also numerous investigations into the energy consumption of applications running in local environment (e.g., desktop systems) without addressing any concern related to the Cloud. Second, we emphasize the scenario of a single application. The studies are excluded if they investigated the energy consumption of a Cloud system or its components (e.g., server, cluster or datacenter [17], [18]) with regarding to some gross workloads from numerous and various applications. Moreover, we also exclude publications that modeled the environmental hardware's energy consumption by notating applications' (or application components') energy consumption (e.g., [19]). This type of studies is not in the context of a single application (component) scenario, either. Third, we are interested in mathematical models instead of those experimental studies that merely compared energy-saving strategies/algorithms without modeling or factor discussions in a generic sense.

Overall, our work synthesizes the related studies that profiled/characterized the energy consumption of applications (even partially) deployed in the Cloud environment. Note that it is acceptable and included if the mathematical models were developed by denoting the energy consumption of environmental hardware (e.g., [20], [21]). In addition, we also select the publications that revealed the changes in energy consumption of a Cloud-based application (or application component) by measuring hardware's energy consumptions either with different workload configurations (e.g., [22]), or with different environmental configurations (e.g., [23]). When it comes to the data transmission energy consumption, in particular, we are concerned with bit/Byte/file data in the application layer rather than the packet/frame in the lower layers of network protocol stack (e.g., [24]).

3 Survey Implementation Methodology

Given the widely accepted SLR guidelines [9], we implemented our survey following a three-stage procedure, namely designing, conducting and reporting. Due to the space limit, we particularly highlight the research questions that essentially drive this literature review, and the inclusion and exclusion criteria that justify our study selection, while only briefly introducing our review conducting process together with the other details.

3.1 Research Questions

During the whole lifecycle of Cloud applications, energy consumption happens mainly when they are being deployed and executed [16]. Moreover, as mentioned previously, the energy for executing a Cloud application is essentially caused by the combined mutual effects between the application software and its environmental infrastructure [10]. Therefore, we decided to summarize the deployment environments and the runtime execution elements of Cloud applications:

RQ1 What deployment environments of Cloud applications have been discussed in the relevant studies?

RQ2 What execution elements of Cloud applications have been discussed in the relevant studies?

Although there is no doubt that running Cloud applications will cause energy consumption, it is more valuable to identify influential factors to understand why different amounts of energy could be consumed even for the same application to achieve the same (or comparable) performance quality. Following the previous research questions, it is natural to distinguish between the environmental factors and the application workload factors:

RQ3 What environmental factors and their influences have been studied in Cloud application energy consumption?

RQ4 What workload factors and their influences have been studied in Cloud application energy consumption?

Through reviewing the modeling studies, one of our main purposes is to reveal Cloud applications' energy consumption models, because the mathematical models can theoretically explain how the energy is consumed:

RQ5 What models have been developed for abstracting the energy consumption of Cloud applications?

3.2 Review Process

By using the quasi-gold standard to manipulate search strings [25], we retrieved over 3,000 publications from the five dominant electronic libraries (namely ACM Digital Library, Google Scholar, IEEE Xplore, ScienceDirect, and SpringerLink), and initially identified 394 studies through quickly scanning their titles and abstracts (note that we only screened the first 50 pages from Google Scholar). In particular, considering that the term "Cloud computing" was coined in 2006 [26], we did not search the literature published before 2006.

After further examining the full texts of the initially collected studies against the inclusion & exclusion criteria, we finally selected 76 papers to fit in this survey. It is notable that we have employed two strategies to reduce the selection bias and improve the fundamental reliability: First, we conducted pilot reviews to try to well establish and polish the inclusion & exclusion criteria in advance. Second, we organized regular meetings to discuss the unsure issues and cross-reviewed the borderline papers.

At last, a data extraction schema was developed to guide paper review and data identification in a structured fashion. In detail, the raw data were gradually extracted from the selected studies and aggregated into a big table to facilitate the overall data synthesis.[1] Based on the data analysis, we deliver the review results and discussions by respectively addressing the aforementioned research questions, as specified in the following section.

4 Review Results and Discussions

4.1 Deployment Environment of Cloud Applications (RQ1)

It has been identified that the deployment environment has significant effects on the energy consumption of Cloud applications [27]. Recall that a Cloud application is generally based on a multi-resource collaboration, and the application tasks could be deployed into different places typically including local devices and Cloud virtual machines [28]. To facilitate locating the energy consumption sources when running Cloud applications, it would be useful to outline a generic deployment architecture in the context of Cloud computing. By extracting the information about deployment configurations from the reviewed studies, we draw an evidence-based environmental architecture for Cloud application deployment, as shown in Fig. 1.

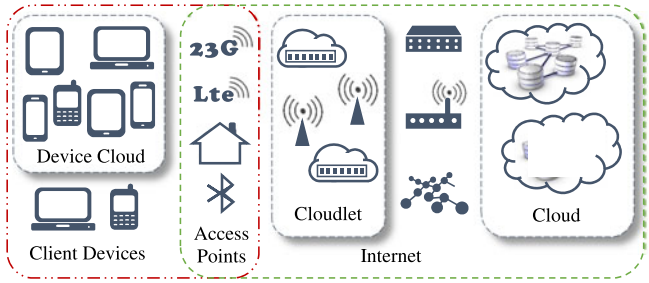

Fig. 1. Evidence-based environmental architecture for deploying Cloud applications.

- *Cloud:* Being located at the far end of the deployment architecture (cf. Fig. 1), the Cloud provides on-demand computing resources for users through the Internet. The Cloud computing paradigm is initially a business model by allowing Cloud consumers to avoid upfront infrastructure costs [29]. Driven by the requirement of energy efficiency in ICT, Cloud computing has acted as a promising solution to the global demand for green computing [1], [30], [31]. Although the data centers in production could continuously use tremendous amounts of electricity [32], the Cloud has been advocated to be more environmentally friendly than local computing systems, for multiple reasons ranging from the improvement of utilization through resource multitenancy to the replacement of high-power local equipment with lightweight client devices [31], [33], [34], [35], [36].
- *Cloudlet:* The emergence of Cloudlet is a crucial evolution in mobile Cloud computing [30]. As the mobile and wearable devices are becoming pervasive, the mobile application market is booming [37]. Many Cloud-based mobile applications require low latencies and high data throughput for their remote interactions and/or workload offloading. However, given the large separation between the local devices and the Cloud, moving computation tasks and transferring data have to go through WAN-scale network hops, which would consequently consume considerable energy and incur unacceptable delay and jitters [38]. To satisfy the resource and performance requirement of mobile applications, a natural approach is to push the Cloud closer to its end users. A Cloudlet can be viewed as a mobile-service-oriented and small-scale data center that is beside the clients or at the inner edge of the Internet. Some empirical studies have shown that, because of smaller round-trip delay, the nearby Cloudlet presents a better offloading option for computation-intensive workloads than the distant Cloud [37], [39].
- *Internet:* Recall that accessing the Cloud/Cloudlet relies on the de facto Internet infrastructure [30], [39], [40], and thus the Internet plays an irreplaceable role in the Cloud ecosystem. According to the telecommunication network design principles, the

1. The schema together with the extracted raw data have been shared online: https://goo.gl/JN8r7W

Internet can be segmented into three main parts including access, metro/edge, and core networks [35], [41], besides the content distribution networks and data centers. Such a segment model has been used to estimate the overall power consumption in the Internet by integrating those individual components [41], [42]. From the application's perspective, however, the calculation of energy for data transportation through the Internet only comprises a small set of involved network equipment (cf. Equation (30) in Section 4.5.4). Therefore, to be aligned with the studies on Cloud applications' energy consumption, we simplify the Internet model to be an equipment combination of switches, routers and various links, plus the Cloudlet and Cloud, as illustrated in Fig. 1.

- *Device Cloud:* Considering the potentially spare computing resources of surrounding devices, peer-device offloading has been proposed as an effective option to share workloads through Bluetooth ad-hoc network [39]. A simulation-based theoretical analysis even showed 63 percent more energy saving than traditional offloading to the Cloud [43]. In addition to the cooperation between peer devices, the paradigm of device Cloud has naturally evolved from the increasing average quantity of mobile devices per user or household, for running an application among a set of cooperative devices [20], [37], [44]. By employing different wireless communication access technologies (e.g., WiFi, 2G/3G, LTE, etc.) and including sensors of various kinds (e.g., GPS, camera sensor, air pollution sensor, etc.), the cooperation among sensor nodes can be extended to a broad range, namely mobile wireless sensor network [43]. As a matter of fact, the latest radio frequency technologies and enhanced processing capability make lightweight wireless sensor nodes also feasible to host sensing applications. Since a sensor is inevitably integrated into a particular electronic equipment (e.g., environmental monitor and vehicle diagnostic board) on the client side (or outer edge of the Internet [40]), we still treat the mobile wireless sensor network as part of the device Cloud paradigm.
- *Client Device:* Although there are various types of client devices, the client-side energy consumption of Cloud applications has been discussed largely with respect to mobile handsets such as smartphones and tablets. In fact, mobile devices nowadays are becoming the primary computing platform and a mandatory part of daily life for many users [37], [45], [46], [47]. Unfortunately, due to the slow development of battery technology compared to the semiconductor technologies [30], [48], the limited battery capacity has been identified to be a major bottleneck of mobile handsets, in contrast to the wall-socket-powered platforms [49], [50], [51]. Moreover, given the high demand for computationally expensive Cloud applications (e.g., the increasingly popular use cases of multimedia streaming), the client devices would further experience a significant increase in the local energy consumption [52], [53], [54]. Correspondingly, the relevant studies are pervasively concerned with workload offloading strategies in mobile Cloud computing, in order to alleviate the suffering from the clients' energy shortage.

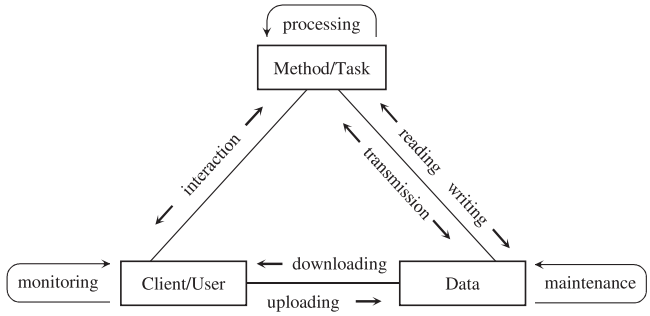

Fig. 2. Evidence-based organization of energy-relevant execution elements of Cloud applications.

4.2 Execution Elements of Cloud Applications (RQ2)

Although there could be an infinite variety in functionality of Cloud applications, we emphasize generic execution elements. To facilitate identifying execution elements of Cloud applications, we pre-list three entities (namely Client/User, Method/Task, and Data) that drive, or are driven by, potential execution elements. At last, nine runtime elements across those entities are extracted from the reviewed papers, as shown in Fig. 2. The discussion about application execution elements is combined into Section 4.2.1.

- *Downloading & Uploading:* In essence, downloading/uploading indicates data access from the user's point of view. We recognize these two activities only when they are specifically discussed in the primary studies, for example the file uploading and downloading from the Cloud [1], [55], [56]. In addition, since the radio frequency module (RF) of mobile devices demands different amounts of energy for sending and receiving data respectively (uploading generally costs more energy than downloading with respect to the same amount of data) [37], [54], we also employ this execution element to cover the separate uplink and downlink wireless transmissions [51], [57].
- *Interaction:* Although the interaction between the client and the remote tasks essentially incurs data exchanging, investigating the energy consumption of interactive workloads could be particularly challenging, due to the fine granularity of communication [11]. Moreover, to intentionally study the mutual actions between a Cloud application and its users, it would be useful to distinguish interaction from the other types of communication elements. For example, instead of reflecting communication data throughput, this execution element is often highlighted when stressing the server load, like user connections [58] and user requests for playing online games [39] or for exploring HTTP websites [22], [59].
- *Maintenance:* If a Cloud application requires data storage, one of its fundamental execution elements would be maintaining the availability and integrity of data. In practice, it is common to spread data across different locations to improve the data accessibility and reduce the likelihood of data loss [3]. Given the limited maintenance scenarios in the selected studies, we roughly identify data files to be stored either in the remote data centers (e.g., when employing storage as a service) or in the local client devices (e.g., when

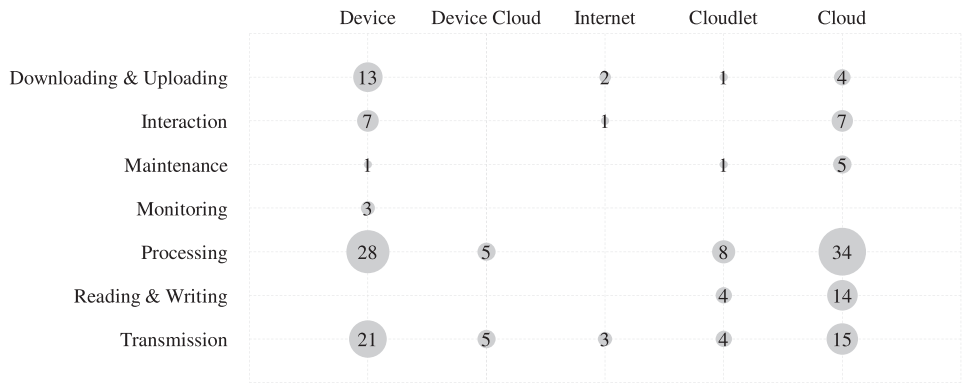

Fig. 3. Distribution of the energy consumption studies with respect to execution elements and deployment environment of Cloud applications.

offloading computational workloads only) [1]. When it comes to the remote data maintenance, storing popular contents in the Cloudlet instead of the Cloud has widely been accepted as an energy-efficient strategy, for reducing the Internet traffic between the content data and their end users [30].

- *Monitoring:* When employing Cloud services, monitoring is one of the primary execution tasks especially in thin-client scenarios [1]. Considering the limited battery capacity of handset devices, runtime monitoring could be a major concern for energy consumption of mobile Cloud applications [49]. Correspondingly, it has been proposed to scale the image frames' backlight levels in particular Cloud applications, like video streaming, in order to reduce the energy consumed in display modules of client devices [53].
- *Processing:* As the name suggests, we treat processing as the processor-centric execution element, such as mathematical calculation (e.g., generating a particular Fibonacci number [60]), logic task execution (e.g., workload-resource scheduling [61], [62]), and data processing (e.g., mapping, shuffling and reducing the input data [2]). Since processor has been considered to be the major power consumer in Cloud computing scenarios [63], processing seems to be the commonest energy-consuming activity that has been discussed in nearly all the selected studies.
- *Reading & Writing:* Compared to data accessing from the user's point of view (i.e., Downloading & Uploading), the application task's perspective considers two types of energy consumption elements of data accessing. The first type focuses on data reading/writing from/to where the data are stored, while the second type emphasizes data transmission through the network. Although not specified in every study, these two element types usually coexist with each other in Cloud applications (e.g., the data fetching requires both disk reading and network transferring [64]). When it comes to Reading & Writing only, one trend is that disk IO is more power-consuming than memory IO, while another trend is that data writing is generally more power-expensive than reading [10].
- *Transmission:* As mentioned above, the element data transmission mainly focuses on application tasks with respect to their data transfer over network resources. Since different tasks of a Cloud application can be executed distributedly, the data transmission could take place not only in the Cloud but also between the Cloud and the client (note that we identify Cloud-client data transmission from a study when it does not emphasize Downloading & Uploading or Interaction). In either case, a Cloud application that transfers large amounts of data would cause a significant proportion of its whole energy consumption, due to two facts: (1) In the Cloud, routers, switches, links and aggregation resources consume more than 30 percent of the total energy [65]; (2) On the client side, data communication has significant impacts on mobile devices' energy consumption [66].

4.2.1 Summary

According to the investigated execution elements and deployment environment of Cloud applications, we distribute the selected studies over a bubble plot, as shown in Fig. 3. It is notable that the same study could have been counted in different bubbles, because one energy investigation might include multiple execution elements and different environmental components (e.g., [1]). With regarding to the execution elements, a clear trend is that most studies have focused on task processing and data transmission, which confirms computation and communication as two major concerns about a Cloud application's energy expense (e.g., using a communication-computation ratio to characterize application workloads and analyze its influence on energy efficiency [67]). Among the environmental components for Cloud application deployment, client devices and Cloud have attracted the most research attentions. By examining their research methods, the reason seems to be twofold: (1) Client devices can directly be controlled and measured; and (2) Cloud data centers can be simplified into a local-server simulation, while the local servers are controllable and measurable. Such a distribution confirms that uncontrollable deployment environment makes addressing a Cloud application's energy consumption more challenging and complex. Correspondingly, by abstracting the uncontrollable aspects, modeling and model-based simulations would be a practical and effective research approach in this case.

4.3 Environmental Factors and Their Influences on Energy Consumption of Cloud Applications (RQ3)

Although the environmental architecture is straightforward (cf. Section 4.1), the deployment of a Cloud application could require sophisticated environmental configurations, and different environmental conditions might in turn drive different

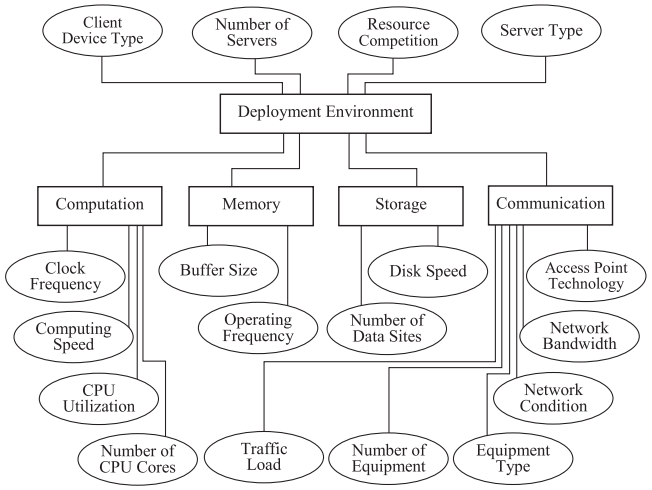

Fig. 4. Evidence-based deconstruction of environmental configurations of Cloud application deployment.

deployment strategies (e.g., the right data distribution with excellent connectivity would be wrong under poor communication channels [49]). In essence, it is the detailed configurations that expose significant environmental impacts on the energy consumption of Cloud applications [59], [68]. To alleviate the complexity in energy analysis with various deployment configurations, it would be valuable to identify individual environmental factors and distinguish their energy influences between each other. Given the fine-grained decomposition of the IT infrastructure [69], the existing studies were mainly concerned with four Cloud resource types, i.e., computation, communication, memory and storage. We accordingly group and report the identified environmental factors, as organized through an entry-relationship diagram in Fig. 4.

4.3.1 Communication Environmental Factors

(1) *Access Point Technology:* Nowadays diverse network technologies are available in different situations for accessing Cloud services, ranging from traditional Ethernet to modern cellular telecommunication. The energy consumption influenced by different technologies is mainly discussed with regarding to client devices [70], [71]. Among the popular access point technologies, WiFi and Ethernet generally consume less energy than cellular wireless networks [30], [35], [48], [49], [72], [73]; although providing lower data rate, Bluetooth could be 80 to 120 percent more energy efficient than WiFi [37]; as for the cellular networks, LTE (4G) consumes more power than UMTS (3G), followed by EDGE (2G) [39], [54].

(2) *Network Bandwidth:* As indicating the maximum channel capacity, the network bandwidth is considered to have a positive impact on reducing both the transmission delay and the energy consumption of Cloud applications [62], [74]. Consequently, bandwidth has become a critical concern for computational offloading in the context of mobile Cloud computing [75]: The offloading effort is not preferred until the connection has sufficient bandwidth, and the benefit of offloading enlarges as the network bandwidth increases [47], [76]. In particular, in addition to the TCP stream bandwidth between different computing resources [21], [77], the researchers are also concerned with the bandwidth of network equipment (e.g., access points [73], [78] and base station [46]).

(3) *Network Condition:* Given the same communication coefficients, better channel quality improves Cloud applications' energy performance [40], while poor network conditions worsens both response time and energy efficiency [45], [79]. The network condition can be reflected by the signal strength or the signal to noise ratio [78]. When the signal strength is low, the relevant network devices will have to increase their power levels for data transmission [39], and will correspondingly end up with higher communication cost [48]. Furthermore, weak signals would lead to high chance of network unavailability [28]. In the worst case, significant energy would be consumed for frequently reestablishing the broken connections, rather than actual data transmission [52].

(4) *Network Equipment Type:* Recall that the Internet topology involves various network equipment, while different types of equipment have different power profiles. Thus, the network equipment types are specified particularly when analyzing the communication energy consumption in Cloud applications [62], [70]. For example, the energy for delivering one bit data through the Internet would be associated with the power consumed in multiple gateways, switches, routers, and high-capacity wavelength division multiplexed fiber links located in different network segments [1], [30], [35].

(5) *Number of Network Equipment:* As mentioned above, a communication line could comprise multiple groups of identical network equipment, and in practice the data traversal would hop through different types of equipment at different amounts [1], [30]. In particular, the number of routers (and their power profiles) was emphasized for the energy expenditure along a data transmission path [64].

(6) *Traffic Load:* Although a network equipment's power profile is predefined by its manufacturer, its practical power consumption would vary depending on the equipment's traffic load [64]. Meanwhile, the traffic load ratio also indicates the resource utilization level of network devices [62]. Similar to the CPU utilization, higher traffic load would increase the communication energy consumption for Cloud applications.

4.3.2 Computation Environmental Factors

(1) *Clock Frequency (and Supply Voltage):* CPU's power consumption is dominantly influenced by its supply voltage [80]. Since the supply voltage is about linearly proportional to the operating clock frequency [62], and only frequency can be altered without making physical adjustments [32], most researchers have mainly focused on the clock frequency as a factor [20], [67], [81], [82], [83], [84], [85]. Intuitively, scheduling low clock frequency will scale down the supply voltage, which eventually brings power saving for CPU [86]. With relax application deadlines, the frequency (or voltage) downscaling has become a preferable approach to energy saving [21], [40], [43], especially for non-CPU intensive workloads [10],

[87], [88], [89], [90], [91]. In particular, fine-grained frequency levels seem to be more energy friendly for Cloud applications [92], [93].

(2) *Computing Speed:* The capacity of a Cloud computational resource can be measured by its computing speed in millions of instructions per second (MIPS) [77]. In general, maintaining high processing speed would consume more energy [40]. In mobile Cloud computing, the speeds of client devices and Cloud servers are usually discussed together, in order to calculate their computing speedup (i.e., the Cloud-client computing speed ratio) [75], [78]. The bigger speedup might indicate the better offloading opportunity, and lead to the higher application performance and the lower energy consumption [46], [47], [74].

(3) *CPU Utilization:* The studies [35], [58] considered the power consumption in a server to be an exponential function of its CPU utilization, and the high CPU utilization is related to the underlying large workload size. Accordingly, higher utilization would result in more energy consumption within the same size of time window [91].

(4) *Number of CPU Cores:* The power consumption of a Cloud computational resource depends on the number of its active CPU cores [94], with a proportional linear relationship [95]. When the physical cores are saturated, adding more workload will not further increase the resources power usage [91]. On the other hand, employing more CPU cores to address the increasing workload will significantly consume more energy due to the increased CPU power and parallelization overhead [90]. Thus, allocating more than enough resources will inevitably result in wastes of energy [95]. Note that utilizing more computational resources to improve a Cloud application's processing concurrency is not a concern here. Multiple factors' combinational impact on energy consumption is discussed in Section 5.

4.3.3 Memory Environmental Factors

(1) *Buffer Size:* As a generally predefined factor, memory buffer size could have to be decided by developers before the Cloud application deployment. The experiments showed that buffering different sizes of data would be sensitively influential on the energy costs of not only the data I/O methods but also the data compression/decompression [10], [23], [82]. For file operations, buffer size between 64 and 256 KB seems to be the most energy-efficient setting [23].

(2) *Operating Frequency:* Memory operating frequency has been viewed as one of the fundamental contributors to the power consumption in memory [91]. Similar to the CPU clock frequency, higher memory frequency will also consume more power.

4.3.4 Storage Environmental Factors

(1) *Disk Speed:* Among all the indexes of a storage device, the disk speed is emphasized in the energy expenditure of an application's storage I/O operations. [64]. The power characteristics of disk speed and other indexes are essentially determined by storage device manufacturers.

(2) *Number of Data Sites:* Spreading data across different sites is a common practice to improve data availability. Correspondingly, for a Cloud application, the more sites need to be visited, the more energy and time will be consumed for more data transmissions [64].

4.3.5 Other Environmental Factors

(1) *Client Device Type:* Although various user handsets do not show big difference in energy consumption for running mobile Cloud applications [54], the client device type indeed matters when making comparison among desktops, laptops and cell phones [20], [70], [71]. Given different power profiles, replacing a personal computer with a low-power consuming device would make the same Cloud application more energy-efficient in a generic sense [35]. If emphasizing the overall share of power consumed in the device communication (e.g., the WiFi interface has a bigger share of the power consumption in smartphones than laptops), however, larger client devices seem preferable for Cloud applications with respect to their energy consumption [96].

(2) *Number of Servers:* In a Cloud host, provisioning more virtual machines could require more physical servers [91], and activating more physical servers implies enhancing the needed power level [67]. Meanwhile, the increased maintenance overhead after provisioning more virtual machines will eventually increase the energy consumption per task in an application [68]. Therefore, selecting a suitable number of servers should optimize the overall power consumption and the total workload [86]. Similar to the aforementioned factor of number of CPU cores, allocating more than enough servers will cause energy waste during the execution of a Cloud application, even if employing sophisticated energy saving mechanisms [97].

(3) *Resource Competition:* If holding the computing resource constant, fierce resource competition could dramatically increase the corresponding energy consumption, no matter what the resource (component) is. For example, configuring more virtual machines within the same physical server will increase the CPU activities and incur extra scheduling overhead [61]. Hosting multiple application instances in a single virtual machine would consume more energy than running application instances separately [27]. As for the resource components, the intense competition for access point connections [78], CPU processes [72], memory footprints [87], and disk IO bandwidth [91] have all been proved negatively impacting Cloud applications' energy efficiency.

(4) *Server Type:* The relevant studies have addressed the types of physical server, virtual server and Web server for their influences on Cloud applications' energy consumption. The physical server type can further be defined by using processor number or types (e.g., Intel versus ARM-based processors) [55], [84]. Given a particular Cloud server pool, the large heterogeneity in server types will result in high variance in the application execution time [67]. As for virtual servers, vertical scaling (adjusting the server type) has clear impacts on the energy consumption and performance of a Cloud application. However,

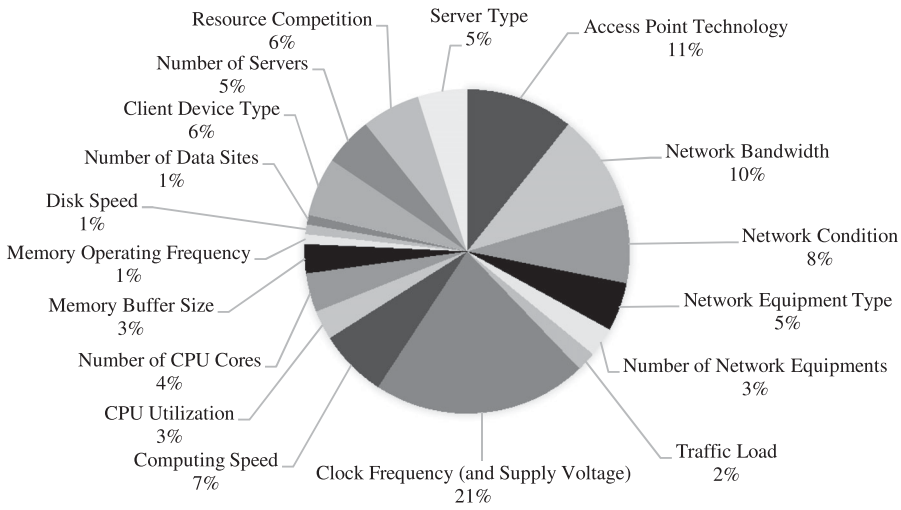

Fig. 5. Study popularity of environmental factors that are influential on energy consumption of Cloud applications. (The total number of factor-studies is 103.)

the specific influences of different virtual machine types are closely related to the application types (workload characteristics) [68]. For example, among different HTTP Web servers, Apache and Lighttp are more energy efficient for lightweight workload, while Nginx consumes relatively less power at faster user arrival speed [22].

4.3.6 Summary

Overall, we have identified 18 environmental factors from the relevant studies. To facilitate tracing back to the reviewed studies, relevant publications are specified for each of the factors. Since the identified factors were not evenly studied, it would be useful to reveal to what extent those factors concerned researchers. Here we employ factor-studies as a metric to measure the popularity of the identified factors, i.e., one factor-study of a particular factor indicates that the factor is involved in one study. The popularity distribution is illustrated in Fig. 5.

It is clear that the CPU clock frequency has been studied as an outstanding environmental factor, followed by the technology of access points and the network bandwidth. As for the factor-study distribution over the four resource types, we only found five studies for two memory factors and one study for two storage factors. This huge imbalance in factor-studies further confirms computation and communication as two major energy concerns in the existing research work from the environmental perspective.

In particular, there are conflict opinions about adjusting CPU clock frequencies for energy saving, particularly through dynamic voltage and frequency scaling (DVFS). Although intelligently scaling frequency can improve energy efficiency, its benefits seem to be trivial [20], and the achievable energy saving could be 13 percent [82] to 20 percent only [84]. Furthermore, different applications might have their best energy efficiency at different optimal frequencies [90], and thus the same DVFS scheduling could only be sub-optimal for those different applications [88].

It is also notable for Access Point Technology that, although WiFi is generally more energy efficient than the cellular technologies, the superiority of WiFi becomes marginal if the utilization of cellular is high (for example when transmitting large bulks of data) [98]. Meanwhile, the efficiency of WiFi in saturation traffic would significantly degrade due to packet loss and retransmissions.

4.4 Workload Factors and Their Influences on Energy Consumption of Cloud Applications (RQ4)

Since the energy for running a Cloud application is tightly coupled with its workload [61], [62], we identify energy-related factors by deconstructing Cloud application workloads. In Cloud environments, an application's workload can be described through one of three different aspects (namely Terminal, Activity, and Object) or a combination of them [69]. Correspondingly, we further organize the workload factors into those three aspects respectively, and use an entry-relationship diagram to illustrate the organization, as shown in Fig. 6. In particular, we consider application type to be an inherent attribute of a Cloud application, and thus "application type" [66], [85], [86], [96] is not regarded as a factor in our survey. In other words, we claim that the type of a Cloud application has already been reflected by its workload characteristics (e.g., the specific communication-computation ratio).

4.4.1 Terminal-Related Factors

The client-side terminals usually act as workload generators in interaction-intensive Cloud applications.

(1) *Number of Clients:* As workload generators, the client-side terminals can be either end users [35] or machines [55], [56], and the number of clients have been used to reflect the size of the generated workload. Naturally, the more number of clients an application serves, the more electric energy the application consumes.

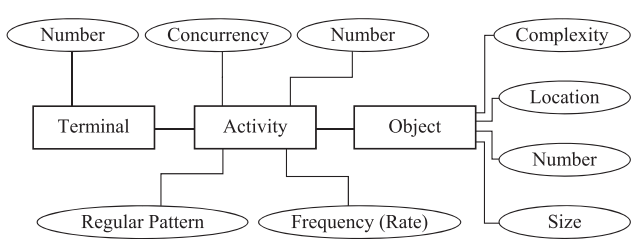

Fig. 6. Evidence-based deconstruction of Cloud application workloads.

4.4.2 Activity-Related Factors

Revoking the previous analysis in Section 4.2, here we identify factors mainly related to those generic application execution elements.

(1) *(Data) Access Pattern:* Data accessing refers to the reading and writing activities. Simple access patterns are relevant to activities only, such as one-time access, repeat access, and cyclic access; while sophisticated access patterns are associated with both the activities and the spatial distance between data locations, such as sequential access, nested access, and random access [82]. When accessing the same amount of data, longer distance traversals will apparently consume more energy. For example, random access has been empirically verified to be significantly more energy expensive [10].

(2) *(Data) Transmission Rate:* Without exceeding physical bandwidths, the power consumed in both servers and network equipment is a proportional function of the total data transmission rate in a Cloud application [35], [55], [56]. However, data transfer at higher bit rate would be more energy efficient (i.e., less energy consumption per bit) [11], and therefore the downloading speed should be set as high as possible to save energy for client devices [51]. On the contrary, the energy consumption per bit was identified to be an increasing function of the data uploading rate from mobile devices. Considering that the low-speed traffic flow's impact on the overall power consumption is generally negligible [35], decreasing the uploading speed has been argued to be an energy optimal solution on the client side (with flexible time limit) [51].

(3) *Number of (User) Connections:* For a Cloud application at runtime, one "connection" indicates an active user session, no matter what activity is issued from the client side. When more user sessions are active, more energy consumption of the application will be incurred [58], [68]. The user connections can be sequential, overlapped, or concurrent (e.g., file downloading from the Cloud [1]). In the concurrent case, more user activities would lead to an increase in Cloud resource usage, and the extra scheduling and synchronizing overhead could in turn increase each user request's processing time [59].

(4) *Processing Concurrency:* Concurrent processing activities commonly exist in parallel applications, and the concurrency can be measured by the amount of processes. Due to the overhead of scheduling, both overall and per-task energy consumption could increase with the number of processes [61], [68]. However, unlike the other types of activities, the concurrency is generally for speeding up workload processing, rather than influencing the workload size. Accordingly, although incurring extra scheduling, increasing the degree of parallelism in a Cloud application can still significantly improve its energy efficiency (i.e., the workload-energy ratio) [84], [87], [90]. In particular, when memory footprints are relatively small, starting multiple processes within less computing resources can be even more energy friendly [95], until reaching the maximum utilization or physical limits of the resources (e.g., the total number of hyperthreads) [91], [94].

(5) *(User/Task) Arrival Rate:* Following the convention of the primary studies, we also use "arrival rate" to represent the frequency of user interactions and task processing. In general, the faster user arrival rate [22] and the shorter inter-arrival time between two consecutive tasks [92] both imply the tenser workload, and correspondingly result in the higher power consumption of a Cloud application. Note that the actual energy consumption eventually depends on the application's execution time, as specified above.

4.4.3 Object-Related Factors

Objects connect, and usually act as targets of, activities in workloads. Similarly, given our previous analysis in Section 4.2, we identify data and task as two types of objects in Cloud applications. In particular, following the object-oriented thinking, a task can be viewed as a composite object (or a dividable piece of workload) that might include other types of workload elements.

(1) *Data Location:* Locality could be a significant contributor to the energy consumption of data accessing. As mentioned in *Data Access Pattern*, it is the data location that essentially impacts different patterns' influences [82]. Thus, moving data closer to where they are needed seems to be an energy saving principle. For example, the collocated data and compute configuration delivers the best energy profile [48], while distributing data and compute nodes into different layers will result in more energy consumption [2].

(2) *Overall Data Size:* The existing studies exhibit a consensus on the positive correlation between the overall data size and the energy consumption of a Cloud application, even though the correlation was studied in various contexts. For instance, the input data size is a major driver behind the computation workload [20], [43]; the energy incurred by accessing activities mainly depends on the data length [64]; and the amount of data to be transmitted is one of the discriminating factors for communication energy cost [50], [62], [66]. In the context of communication, the relevant studies further distinguish between two scenes: The first is on the traffic volumes exchanged between the client and the Cloud [1], [35], [37], [45], [47], [48], [54], [55], [56], [70], [71], [72], [74], [78], [84], while the second is on the data segments involved in, and transferred between, individual application tasks [2], [21], [46], [61], [68], [73], [77].

(3) *Transactional Data Size:* Although the required energy increases proportionally to the overall data size, small-data transactions in a Cloud application show a negative correlation with the energy consumption. In practice, the data block per transaction can vary from several bytes to multiple megabytes [82]. Given the same amount of data in an application, dealing with smaller-data-size transactions would cause longer execution time and higher energy expense [68]. Consequently, packing a set of small data requests into a bulk transaction becomes an effective approach to improve the application's energy efficiency [11], [82]. Note that the aforementioned data segments involved in application tasks do not

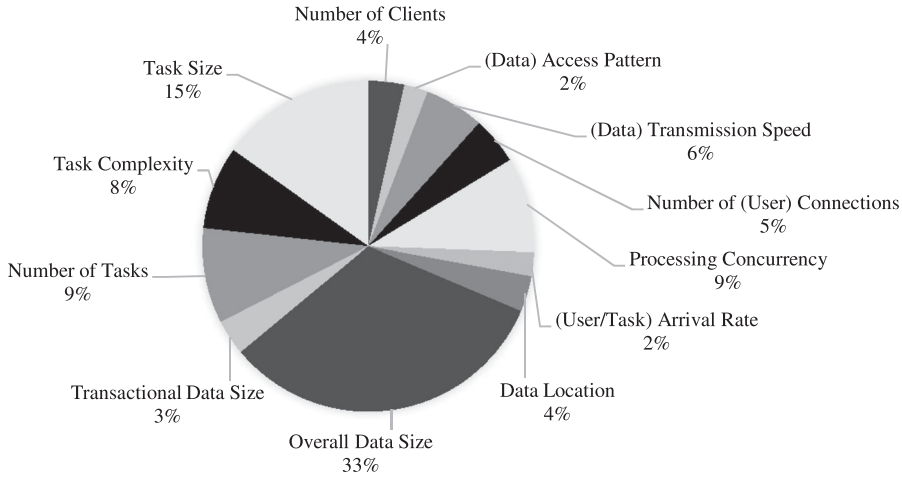

Fig. 7. Study popularity of workload factors that are influential on energy consumption of Cloud applications. (The total number of factor-studies is 86.)

necessarily act as transactional data pieces, because a task might further comprise numerous transactions.

(4) *Number of Tasks:* By representing Cloud applications as task interaction graphs (e.g., directed acyclic graph), the number of task nodes and edges has been used to reflect the whole workload (i.e., graph size) [21], [28], [67], [92], [93], [97]. Since more tasks usually imply more data and more application activities at runtime, the corresponding application execution will inevitably require more energy [64]. Moreover, considering the extra overhead and energy for task scheduling, a larger number of tasks in a Cloud application will lead to higher average energy consumption per task [68].

(5) *Task Complexity:* The computational complexity in tasks or functional modules is closely associated with the Cloud application's energy consumption [35], [72], [99], as complex computation requires more computing resources and/or causes longer execution time. To verify this association, the empirical studies varied task complexity mainly through topping up functions [58] and increasing the load of mathematical calculations [60], [84], while the simulation study [43] characterized the complexity in computation algorithm as a random variable with Gamma distribution.

(6) *Task Size:* As mentioned previously, a composite-object task can further be defined as a combination of the input/output data and computation workload [46], and therefore the size of a task can partially be reflected by the data size [61] or together with the computation complexity [43]. To avoid duplication, we only focus on the amount of computation workload [71] that has been widely depicted as the number of CPU cycles [21], floating-point operations [37], [78], and processing instructions [46], [47], [73], [77], [92]. In fact, the CPU cycles of a computation task have been treated as a linear function of the data input the task [20], and the computation complexity can also be translated into particular number of instructions [50].

4.4.4 Summary

As listed above, we have identified 12 workload factors in total. In a similar fashion to Section 4.3.6, we also use numerical factor-studies to reflect to what extent different environmental factors have concerned researchers, as illustrated in Fig. 7. It is again notable that popular factors do not necessarily act as main contributors to energy consumption.

By isolating individual factors' impacts on energy consumption from each other, the sizes of data and task (the processing workload) seem to be the main energy-related factors in a Cloud application. In fact, there has been a wide consensus on these two factors among the literature and reality: Computational tasks rely on the major power consumer of computing resources [63], while the data lead to communication and storage costs. Such a factor concentration roughly matches the main environmental factors (cf. Section 4.3.6) in terms of their potential interactions (i.e., task processing and data communication).

Since Cloud application workload is usually reflected by a combination of factors, in practice, one factor's influence on energy consumption could be correlated with or even constrained by others. For example, task size and task complexity can sometimes interchangeably indicate each other ([43] versus [50]); the number of tasks and data size are frequently used together to represent the overall workload size (e.g., [78]); while the degree of parallelism in a Cloud application also depends on the resource allocations (e.g., [90]). We leave more discussions about combinational influences of factors to Section 5.

4.5 Energy Consumption Models of Cloud Applications (RQ5)

Recall that it is extremely challenging to deal with energy-related issues of Cloud applications due to the inherent complexity in the applications themselves and the heterogeneity in their deployment environments [59]. By abstracting energy consumption behaviors and details, the mathematical models have pervasively been employed to help understand and in turn investigate how the energy is consumed for running a Cloud application. To facilitate discussing, comparing, and reporting the identified energy consumption models, we unify the various notations from the relevant studies, as listed in Table 1. Moreover, except for the models that hold implicit environmental views, we follow the previous environmental deconstruction to organize the identified models respectively representing overall energy consumption as well as computation, communication, and storage energy consumption of Cloud applications. In particular, we did not find memory-specific energy consumption models in the context of Cloud applications.

TABLE 1
Summary of Key Notations

Symbol	Brief Explanation
a, k	Predefined constant coefficients.
A	The Cloud application.
C	Total CPU cycles as the computational workload involved in a particular task.
$D(\cdot)$	Data size function either of Cloud application/tasks (e.g., $D(n_i)$), or of environmental resource/items (e.g., $D(r_i \rightarrow)$). In the former case, it represents the size of data involved in an application/task. In the latter case, it uses $\rightarrow$ or $\leftarrow$ to indicate the size of data sent/received from/by a resource item.
$\widehat{D}(\cdot)$	The maximum content capacity (or size) of the memory or the hard disk.
$e(\cdot)$	Workload-oriented energy rate function of Cloud application/tasks (e.g., $e(n_i)$). Unlike $P(\cdot)$, it defines the energy expense during a unit of time when dealing with workloads.
$E(\cdot)$	Energy consumption function of Cloud application/tasks (e.g., $E(A)$). It can use a superscript to specify the relevant resource (e.g., $E^{client}(\cdot)$), and a subscript to indicate the energy consumption component (e.g., $E_{active}(\cdot)$).
f, v	The operating frequency (i.e., f) and supply voltage (i.e., v).
M, N	The total number of environmental resource items (i.e., M) and Cloud application tasks (i.e., N).
n_i	The ith task of the Cloud application A. In particular, the subscript can be replaced with cpu, net, mem, or $disk$ to indicate a particular type of resource-intensive task.
$P(\cdot)$	Power consumption function of environmental resource/items (e.g., $P(r_i)$). It can further include t or $\Phi(r_i)$ to indicate the power at a particular time point or data throughput (e.g., $P(r_i, t)$ or $P(r_i, \Phi(r_i))$). If needed, a subscript is used to specify the power consumption component (e.g., $P_{idle}(\cdot)$).
r_i	The ith resource item in a particular resource pool $R(\cdot)$. If needed, a particular resource and/or its component can further be specified in the subscript (e.g., $r_{client,cpu}$).
$R(\cdot)$	Environmental resource function of Cloud application/tasks (e.g., $R(A)$).
$S(\cdot)$	Compute speed function of environmental resource/items (e.g., $S(r_i)$).
t	A particular time point.
$T(\cdot)$	Time span function either of Cloud application/tasks (e.g., $T(A)$), or of environmental resource/items (e.g., $T(r_i)$). In the latter case, it can use a subscript to imply the resource state (e.g., $T_{idle}(\cdot)$).
$U(t)$	Resource utilization ratio at time point t.
$W(\cdot)$	Workload size function of Cloud application/tasks (e.g., $W(A)$). It can further include t to indicate the workload at a particular time point (e.g., $W(n_i, t)$).
$\alpha, \beta, \gamma, \lambda$	Regression parameters that need to be determined by experimental measurements.
δ	A particular fraction ratio.
Θ	Data transmission channel quality with variable value $0 < \Theta < 1$.
τ	Execution time of a particular task at the maximum processing capacity.
$\Phi(\cdot)$	Data throughput function of the channel (either network communication or data reading/writing) between two resource items (e.g., $D(r_i \rightarrow r_j)$). If needed, it is possible to emphasize one resource item only (e.g., $\Phi(r_i \leftarrow)$), and also to ignore the data flow direction (e.g., $\Phi(r_i)$).
$\widehat{\Phi}(\cdot)$	The maximum data throughput capacity (or bandwidth) between two resource items or of a single resource item.
Ω	The set of power-consuming components contained in a particular resource item.

4.5.1 Environment-Implicit Energy Consumption Model

As the name suggests, the environment-implicit energy consumption models are purely based on the analysis of Cloud applications, with little consideration of the deployment environment. Without loss of generality, we exploit the widely employed directed acyclic graph (DAG) as a generic model of Cloud application A in our discussion, as shown in

$$A : \begin{cases} DAG = \{Node, Edge\} \\ Node = \{n_i | 1 \leq i \leq N\} \\ Edge = \{(n_i, n_j) | n_i \in Node, n_j \in Node\}, \end{cases} \quad (1)$$

where the application's DAG comprises N nodes and at most $N \times N$ edges. By partitioning A into functional pieces, each node n_i indicates a workload task, while each edge $(n_i,\ n_j)$ represents the precedence constraint between two consecutive tasks. Unlike the application modeling in [75], [99], we treat data transmission as a workload task represented by a node instead of an edge.

By focusing only on the execution duration and the required energy unit of each workload task, the most straightforward energy consumption model of A was given in [77], [100]

$$E(A) = \sum_{i=1}^{N} e(n_i) \cdot T(n_i), \quad (2)$$

where $E(\cdot)$ represents a generic energy consumption function, while $T(\cdot)$ is a generic makespan function. Note that $e(n_i)$ is the energy unit consumed by the task n_i during a unit of time, which essentially is a workload-oriented notation [50], [61], [68] in contrast to the power consumption in environmental resources. In addition to the task energy per time unit, there are also other types of workload-oriented energy units, e.g., energy per user or energy per bit [35].

When individual workload tasks have the same functionality, they can be grouped together to facilitate energy consumption modeling. For example, in the context of a MapReduce workflow, there are generally mapping, shuffling and reducing tasks. Correspondingly, the study [101] defined a *function-group-based energy consumption model* as

$$E(A) = \begin{cases} E(map) + E(hold) + E(reduce) & \text{if local data,} \\ E(map) + E(replicate) + E(hold) + E(reduce) & \text{if distributed data.} \end{cases} \quad (3)$$

Recall that there are mainly four types of infrastructural resources (cf. Section 4.3). Without necessarily knowing the

environmental details, similarly, we can also group the tasks that are related to the same resource-intensive workload. As for the task interactions, their energy consumption comprises an integration of task computation and information communication between tasks [62]. Although few modeling studies were concerned with the four resource types simultaneously, we summarize such a *resource-group-based energy consumption model* inspired by the empirical investigation [91], as shown below

$$E(A) = E(commun.) + E(comput.) + E(memory) + E(storage). \tag{4}$$

Since this model is inherently associated with Cloud applications' deployment environment, we further treat it as a bridge between the environment-implicit and the following environment-specific energy consumption models.

4.5.2 Environment-Specific Overall Energy Consumption Model

When it comes to the environment of a Cloud application, we are only concerned with the IT equipment, while not including cooling and other facilities. From the viewpoint of resource partitioning, the deployment environment of Cloud applications has normally been modeled as a resource pool comprising a set of resource items

$$R(A) = \{r_i \mid 1 \leq i \leq M\}, \tag{5}$$

where r_i is the ith resource item within the pool $R(A)$ consisting of M resource items. Since we focus on the environmental resources with respect to a single Cloud application, in this survey, we clarify that $R(A)$ is only composed of the resource items employed by the aforementioned Cloud application A. Moreover, the employed resource items might have different types [102], and the same type of resource items are not necessarily identical [85]. Then, the energy consumption of A can be modeled based on the involved resources' power consumptions. In fact, a key characteristic of environment-specific modeling is that it relies on the power consumption of environmental resources. For example, by denoting the power consumed in the resource item r_i at time t to be $P(r_i, t)$, the studies [87], [90] modeled the energy expense of a parallel application A running with M resource items during time interval (t_1, t_2)

$$\Delta E(A) = \sum_{i=1}^{M} \int_{t_1}^{t_2} P(r_i, t) \cdot dt. \tag{6}$$

If we define every resource item to be a combination of various power-consuming components, $P(r_i, t)$ of resource r_i can further be specified into $\sum_{j \in \Omega} P(r_{i,j}, t)$, where Ω is the set of power-consuming components [102]. By dividing Ω into the aforementioned four resource types (namely *cpu*, *net*, *mem* and *disk* for short), we are able to update Equation (6) and make it compatible with Equation (4)

$$\begin{aligned}\Delta E(A) = \sum_{i=1}^{M} \int_{t_1}^{t_2} \sum_{j \in \Omega} P(r_{i,j}, t) \cdot dt = \sum_{i=1}^{M} \int_{t_1}^{t_2} \Big(P(r_{i,cpu}, t) \\ + P(r_{i,net}, t) + P(r_{i,mem}, t) + P(r_{i,disk}, t) \Big) \cdot dt.\end{aligned} \tag{7}$$

If focusing on the CMOS circuits involved in the IT resources [93], since a CMOS circuit has two power consumption components (namely static power and dynamic power), a Cloud application's energy consumption can be distinguished between the static and dynamic parts [28], as shown in

$$\begin{aligned}E(A) &= E_{static}(A) + E_{dynamic}(A) \\ &= \big(P_{static}(R(A)) + P_{dynamic}(R(A))\big) \cdot T(A),\end{aligned} \tag{8}$$

where $P_{static}(R(A))$ and $P_{dynamic}(R(A))$ represent the average static and dynamic power consumed in the application environment $R(A)$ during the application runtime $T(A)$.

In theory, *Static Power* indicates the essential power for keeping IT resources in the power-on state (e.g., maintaining the basic circuits and system clock), which is independent of any workload [21] and cannot be avoided until the whole system is turned off [102], [103]. As such, the static power consumption is normally modeled as a constant without scaling with other factors [86]. In practice, the reverse-bias leakage between diffused regions and the substrate will also result in a particular amount of static power consumption, while the leakage can be proportionally influenced by the temperature [104]. Further considering the proportional impact of dynamic power on the temperature, some studies estimated the static power as a fraction of its dynamic counterpart, and the fraction is usually less than 30 percent [89], [92]. Thus, during the execution of an application, the static energy consumption can be expressed as

$$E_{static}(A) = \delta \cdot E_{dynamic}(A),\ 0\% < \delta < 30\%. \tag{9}$$

Dynamic Power is the dynamic utilization of power in the environmental IT resources when dealing with workloads. Since the dynamic power dominates the whole power consumption in the popular CMOS technology [103], most of the relevant studies only employed the dynamic power for modeling the energy consumption of Cloud applications (e.g., [21], [85]).

Furthermore, from the perspective of a system rather than of a CMOS gate, we distinguish between the active and idle power consumption according to different load levels of a particular IT resource during the execution of a Cloud application [49], [97]. *Active Power* refers to the power for actively executing tasks on an IT resource (i.e., > 0 percent load), and *Idle Power* indicates the power consumption when the IT resource is ready to work while doing nothing (i.e., 0 percent load). Note that IT resources are not truly static at idle states, because there are still backend workloads.[2] To be aligned with the definition of dynamic power (when dealing with workloads), we clarify that static power is excluded when discussing active power and idle power in this survey. In fact, the study [21] has combined idle power with static power (e.g., the power corresponding to the *sleep* resource state [67], [96]) into the so-called *standby power*. Therefore, by focusing on the dynamic power, the dynamic energy expense for completing the Cloud application A can be modeled as

$$\begin{aligned}&E_{dynamic}(A) = E_{idle}(A) + E_{active}(A) \\ &= P_{idle}(R(A)) \cdot T_{idle}(R(A)) + P_{active}(R(A)) \cdot T_{active}(R(A)),\end{aligned} \tag{10}$$

where $T_{idle}(R(A))$ and $T_{active}(R(A))$ respectively indicate the average idle and the average active time of the environmental IT resources $R(A)$. It is noteworthy that $T_{idle}(R(A)) + T_{active}(R(A)) \neq T(A)$. Since different resource items are

2. https://wiki.mcs.anl.gov/cqos/index.php?title=Power_Specifications_and_Model

possible to be alternatively idle during the continuous execution of the Cloud application A, it is improper to use fractions of $T(A)$ to calculate A's idle and active energy consumption.

When it comes to $E_{active}(A)$, one of the active energy components reflects the energy used for driving the data flow of the Cloud application A. The data flow might comprise various interactive execution elements (cf. Fig. 2) with respect not only to network equipment (e.g., [21]) but also to other types of resources (e.g., [102]). From the perspective of a single resource item r_i, the corresponding data flow can be distinguished as either data input or data output. By emphasizing the input/output channel between two consecutive resource items, the energy consumption of A's data flow has been modeled as follows:

$$E_{flow}(A) = \sum_{r_i, r_j \in R(A)} \big(P_{out}(r_i) + P_{in}(r_j)\big) \cdot \frac{D(r_i \rightarrow r_j)}{\Phi(r_i \rightarrow r_j)},\ i \neq j, \quad (11)$$

where $P_{out}(r_i)$ (resp. $P_{in}(r_j)$) is the power of resource r_i (resp. r_j) when outputting/inputting the data $D(r_i \rightarrow r_j)$, and $\Phi(r_i \rightarrow r_j)$ refers to the data throughput between those two different resource items r_i and r_j.

Instead of emphasizing the input/output channel, Equation (11) has been rewritten in [20] by focusing on the input/output activities of individual resource items:

$$E_{flow}(A) = \sum_{r_i \in R(A)} \left(P_{out}(r_i) \cdot \frac{D(r_i \rightarrow)}{\Phi(r_i \rightarrow)} + P_{in}(r_i) \cdot \frac{D(r_i \leftarrow)}{\Phi(r_i \leftarrow)} \right), \quad (12)$$

where $D(r_i \rightarrow)/D(r_i \leftarrow)$ represents the size of output/input data of the resource item r_i, and $\Phi(r_i \rightarrow)/\Phi(r_i \leftarrow)$ indicates the data throughput when r_i is outputting/inputting data.

4.5.3 Environment-Specific Computation Energy Consumption Model

Following the convention of Equations (2) and (5), here we consider a computation-intensive task n_{cpu} running on the compute resource r_{cpu}. As mentioned above, the dynamic power dominates the power consumption of CPU's CMOS circuits, and the dynamic CPU power generally depends on the supply voltage and operating frequency via relation $P_{dynamic}(r_{cpu}) = k \cdot v^2 \cdot f$ [80]. The operating frequency-based model specified in Equation (13) has widely been used for applications' energy consumption in both client devices and Cloud servers

$$E_{dynamic}(n_{cpu}) = k \cdot v^2 \cdot f \cdot T(n_{cpu}) = k \cdot a^2 \cdot f^3 \cdot T(n_{cpu}), \quad (13)$$

where the energy coefficient k depends on the CPU's chip architecture; the linearly proportional relationship between the operating clock frequency f and the supply voltage v is modeled as $v = af$; while a is a constant coefficient.

It is evident that the consumed energy of the task n_{cpu} is directly proportional to its makespan, i.e., $E(n_{cpu}) \propto T(n_{cpu})$ [72]. However, a task's makespan varies in practice due to the dynamic changes in CPU capacity caused by possible voltage scaling at runtime. If using τ to denote the time for executing the task n_{cpu} at the maximum processing capacity, then the practical execution time $T(n_{cpu})$ would be $\tau \cdot \frac{v_{max}}{v}$ [80] or $\tau \cdot \frac{f_{max}}{f}$ [83]. In particular, the levels of voltage v and frequency f are within range $[v_{min}, v_{max}]$ and $[f_{min}, f_{max}]$ respectively. Accordingly, the previous frequency-based energy consumption model has been updated by [83] into

$$E_{dynamic}(n_{cpu}) = \int_0^{\tau \cdot \frac{f_{max}}{f}} k \cdot a^2 \cdot f^3 \cdot dt = k \cdot a^2 \cdot f_{max} \cdot f^2 \cdot \tau. \quad (14)$$

Recall that the computation workload induced by a task can be measured by CPU cycles (cf. Task Size in Section 4.4). Suppose the task n_{cpu} comprises C cycles in total. Its makespan can directly be calculated as C/f at frequency f. Then, as proposed in [20], [38], [92], the energy consumption of such a task can be modeled as

$$E_{dynamic}(n_{cpu}) = k \cdot a^2 \cdot f^3 \cdot \frac{C}{f} = k \cdot a^2 \cdot f^2 \cdot C. \quad (15)$$

In the extreme case, the operating frequency is assumed changeable after every single CPU cycle [40], [43]. Given the single cycle time $1/f_c$ at frequency f_c, one CPU cycle's energy consumption can be represented as $E(cycle) = k \cdot a^2 \cdot f_c^3 \cdot \frac{1}{f_c} = k \cdot a^2 \cdot f_c^2$, and thus the task's energy consumption can be expressed as

$$E_{dynamic}(n_{cpu}) = \sum_{c=1}^{C} k \cdot a^2 \cdot f_c^2. \quad (16)$$

Considering that $f_c \in [f_{min}, f_{max}]$ and there are only limited frequency levels within $[f_{min}, f_{max}]$, we can categorize the CPU cycles into different frequency level groups. By using δ_f to denote the execution fraction of the task n_{cpu} at the frequency f [21], the energy consumption model can be rewritten with regards to either the CPU cycles fractions (i.e., $C \cdot \delta_f$) or the execution time fractions (i.e., $T(n_{cpu}) \cdot \delta_f$), as shown below:

$$\begin{aligned} E_{dynamic}(n_{cpu}) &= \sum_{f \in [f_{min}, f_{max}]} k \cdot a^2 \cdot f^2 \cdot C \cdot \delta_f \\ &= \sum_{f \in [f_{min}, f_{max}]} k \cdot a^2 \cdot f^3 \cdot T(n_{cpu}) \cdot \delta_f. \end{aligned} \quad (17)$$

As explained in Equation (10), the idle state of compute resources caused by a task is generally unavoidable due to workload offloading or imbalanced parallel execution. In particular, a compute resource is considered to be idle when its operating frequency (or supply voltage) reaches the lowest level f_{min} (or v_{min}) [93]. Accordingly, by focusing on the dynamic power, the dynamic energy expense for running the task n_{cpu} on the resource r_{cpu} can be separated and modeled as follows:

$$\begin{cases} E_{idle}(n_{cpu}) = k \cdot a^2 \cdot f_{min}^3 \cdot T_{idle}(r_{cpu}) \\ E_{active}(n_{cpu}) = \sum_{f \in (f_{min}, f_{max}]} k \cdot a^2 \cdot f^3 \cdot T_{active}(r_{cpu}) \cdot \delta_f. \end{cases} \quad (18)$$

Similar to Equations (6) and (7), it is also common to model Cloud application energy consumption without specifying the power details such as operating frequency. For example, by assuming the resource power $P(r_{cpu})$ and the compute speed $S(r_{cpu})$ to be constant when running the task n_{cpu}, the consumed energy was calculated in [73] through

$$E_{active}(n_{cpu}) = P_{active}(r_{cpu}) \cdot \frac{W(n_{cpu})}{S(r_{cpu})}, \quad (19)$$

where $W(\cdot)$ is a generic workload function, and then $W(n_{cpu})$ refers to the workload of the task n_{cpu}. It is clear that the idle state of compute resource has been excluded in this case. Therefore, we particularly label Equation (19) as an active energy consumption model.

Instead of a constant value, the power consumed in a compute resource has been identified to be an exponential function of the resource utilization [58]. By using $P_{idle}(r_{cpu})$ and $P_{full}(r_{cpu})$ to respectively represent the compute resource's empty and full load powers, the energy consumption for running the task n_{cpu} on the compute resource can be modeled as

$$E_{dynamic}(n_{cpu}) = \int_0^{T(n_{cpu})} \Big(P_{idle}(r_{cpu}) + \big(P_{full}(r_{cpu}) - P_{idle}(r_{cpu})\big) \cdot \alpha \cdot U(t)^{\beta}\Big) \cdot dt, \tag{20}$$

where α and β are resource-specific parameters that need to be determined through empirical measurements. The context-dependent notation $U(t)$ denotes the utilization of compute resource at time t. In the straightforward case, $U(t)$ directly equals to the CPU load fraction [96]. As for a multi-CPU server, $U(t)$ was estimated as the number of active CPU cores among all the available ones [94]. Considering that the compute resource utilization would also be proportional to the workload being dealt with, the study [58] further modeled $U(t) = \gamma \cdot W(n_{cpu}, t) + \lambda$, where γ and λ are both resource-specific parameters, and the workload $W(n_{cpu}, t)$ was measured by the number of user connections at time t.

4.5.4 Communication Energy Consumption Model

Similarly, we define a communication-intensive task n_{net} of A to facilitate our discussion. As explained in Section 4.5.2, the task n_{net} can be thought of as a data flow across the involved resource pool $R(n_{net})$, and then $E(n_{net})$ can directly be derived from Equations (11) and (12) [20], [21].

Given the generic architecture for physical environment of Cloud applications (cf. Section 4.1), the resource items can be grouped into Client, Internet, Cloudlet, and Cloud resources. Accordingly, the communication energy consumption of a Cloud application can roughly be divided into four parts [30], as modeled as follows:

$$E(n_{net}) = E^{client}(n_{net}) + E^{internet}(n_{net}) + E^{cloudlet}(n_{net}) + E^{cloud}(n_{net}). \tag{21}$$

It is noteworthy that Equations (11) and (12) are still valid and can be reused for each of the four energy parts by adapting the resource pool.

As the most controllable part, the client side attracts most of the research efforts on modeling communication energy consumption. By treating a client device as a single resource item, a straightforward approach is to follow Equation (12) to estimate the communication energy consumed in client devices, as follows:

$$E^{client}(n_{net}) = P_{send}(r_{client,net}) \cdot \frac{D(r_{client}\rightarrow)}{\Phi(r_{client}\rightarrow)} + P_{receive}(r_{client,net}) \cdot \frac{D(r_{client}\leftarrow)}{\Phi(r_{client}\leftarrow)}. \tag{22}$$

Without distinguishing the power [47] and data [73] between sending and receiving, Equation (22) can be simplified to

$$E^{client}(n_{net}) = P(r_{client,net}) \cdot \frac{D(n_{net})}{\Phi(r_{client})} \quad or \quad E^{client}(n_{net}) = P(r_{client,net}) \cdot \frac{2 \cdot D(n_{net})}{\Phi(r_{client})}, \tag{23}$$

where $P(r_{client,net})$ and $\Phi(r_{client})$ are respectively the transmission power and data throughput of the client device r_{client}. Note that we use $r_{client,net}$ to emphasize the power consumed in the network component of the resource r_{client}; and the notation $\Phi(r_{client})$ completely ignores the data transmission directions. As such, the first expression in Euqation (23) views $D(n_{net})$ as the overall roundtrip data in the task n_{net}, while in the second expression $D(n_{net})$ is doubled to imply the data transmission along both directions.

Considering the influence of uncertain channel quality (e.g., transmission errors), the factor Network Condition (cf. Section 4.3) was introduced to the previous cases [39]

$$E^{client}(n_{net}) = P(r_{client,net}) \cdot \left(\frac{D(r_{client}\rightarrow)}{\Phi(r_{client}\rightarrow)} + \beta_1 + \frac{D(r_{client}\leftarrow)}{\Phi(r_{client}\leftarrow)} + \beta_2\right), \tag{24}$$

where β_1 and β_2 are the channel condition parameters for sending and receiving data respectively, and their values are required to be tested by the client device r_{client} itself [39]. We note that, in this model, the data sending and receiving power of r_{client} are assumed to be identical. By using regression analysis and Wolfram Mathematica, the study [45] even ignored the data transmission power, and proposed the following energy consumption model:

$$E^{client}(n_{net}) = \frac{\alpha \cdot D(n_{net}) - \beta}{\Phi(r_{client})}, \tag{25}$$

where α and β are constant parameters that need to be determined through experimental measurements. Resorting to the Shannon Formula, $\Phi(r_{client})$ was further modeled as $\Phi(r_{client}) = \frac{\widehat{\Phi}(r_{ap})}{number\ of\ clients} \cdot \log_2(1 + \frac{SNR}{Distance(r_{client}, r_{ap})^2})$, with regarding to the signal to noise ratio SNR, the bandwidth $\widehat{\Phi}(r_{ap})$ and resource competition of the access point r_{ap}, and the distance between r_{ap} and r_{client} [78].

By replacing transmission throughput with channel quality, the studies [40], [43] proposed the following convex monomial function to describe the energy used to transmit $D(n_{net})$ bits of data

$$E^{client}(n_{net}) = \gamma \cdot \frac{D(n_{net})^o}{\Theta}, \tag{26}$$

where γ denotes the energy coefficient in the order of less than 10^{-2}, Θ represents the channel state with variable value $0 < \Theta < 1$ at different time slots, and o refers to the order of monomial that depends on the transmission scheduling policy. For instance, the one-shot policy $o = 1$ is used to indicate that the channel state has the biggest influence on the data transmission, and the transmission is finished in one time slot only.

Without conflicting with such a one-shot policy, a further simplified model proposed a directly proportional relation between the energy consumption of a communication task and its data size, i.e., $E^{client}(n_{net}) \propto D(n_{net})$ [54], [72], as shown below:

$$E^{client}(n_{net}) = \lambda \cdot D(n_{net}), \tag{27}$$

where λ is a linear or quantile regression parameter that can be related to the employed access point technology [54].

By analogy with CMOS concern, the network power of client devices, $E^{client}(n_{net})$, can also be separated into static part

and dynamic part [28], where the dynamic part covers the idle and active states [96]. In particular, the active energy for wireless communication between the mobile device's RF module and different access points (cellular versus WiFi) was emphasized by [38], [57], as modeled below. To save space, here we replace the task n_{net} with a dot

$$E^{client}_{active}(\cdot) = \begin{cases} E^{RF}_{ramp}(\cdot) + E^{RF}_{transmit}(\cdot) + E^{RF}_{hold}(\cdot) + E^{RF}_{tail}(\cdot) & \text{if cellular,} \\ E^{RF}_{scan}(\cdot) + E^{RF}_{transmit}(\cdot) + E^{RF}_{hold}(\cdot) & \text{if WiFi.} \end{cases} \tag{28}$$

where $E^{RF}_{ramp}(\cdot)$ refers to the extra energy for switching the RF circuitries from low- to high-power states before the initiation of cellular data transmission; $E^{RF}_{tail}(\cdot)$ indicates the tail energy of high-power duration after the cellular data transmission ends; $E^{RF}_{scan}(\cdot)$ represents the energy for scanning and associating to an available WiFi access point; $E^{RF}_{transmit}(\cdot)$ includes both the uplink and the downlink data transmission energy [46] that can be calculated through Equation (22), and the power value and data throughput need to be adapted to the chosen access point technology; while $E^{RF}_{hold}(\cdot)$ is the energy for keeping the access point interface active during the data transmissions.

Besides the client-side wireless network, the Internet was studied as another communication part for mobile Cloud applications in [62]. The communication energy consumed in the Internet was identified to be relative to the data size, the traffic load ratio and the transmission delay. However, the negative correlation between the transmission delay and the corresponding energy consumption conflicts with the other relevant studies and seems to be incorrect, thus our survey does not include the model proposed in [62].

By focusing on the routers only in the network path of a Cloud application, the study [64] simplified the Internet architecture, and used the number of routers and their power profiles to model the data transmission energy

$$E^{internet}(n_{net}) = \sum_{r_{router} \in R(n_{net})} P(r_{router}, \Phi(r_{router})) \cdot \frac{D(n_{net})}{\Phi(r_{router})}, \tag{29}$$

where $P(r_{router}, \Phi(r_{router}))$ represents the power of the router r_{router} at the data throughput $\Phi(r_{router})$, which implies that the router's power varies depending on its traffic load.

In practice, given different network segments of the Internet, the routers can be specified and classified according to their functions and locations, such as broadband gateway routers and edge/core routers. Moreover, the network path of a Cloud application also includes other types of network facilities like Ethernet switches and WDM transport equipment [35]. In detail, the user traffic over the Internet has been assumed to generally require three hops (over two switches, one broadband gateway router, and one edge router) before reaching the core network, and eight hops (over eight WDM links across nine core routers) within the core network [1]

$$E^{internet}(n_{net}) = 4 \cdot \left(\frac{2 \cdot P(r_{switch})}{\widehat{\Phi}(r_{switch})} + \frac{P(r_{broad})}{\widehat{\Phi}(r_{broad})} + \frac{P(r_{edge})}{\widehat{\Phi}(r_{edge})} + \frac{2 \cdot 9 \cdot P(r_{core})}{\widehat{\Phi}(r_{core})} + \frac{8 \cdot P(r_{wdm})}{2 \cdot \widehat{\Phi}(r_{wdm})} \right) \cdot D(n_{net}), \tag{30}$$

where $P(r_{switch})$, $P(r_{broad})$, $P(r_{edge})$, $P(r_{core})$, and $P(r_{wdm})$ refers to the powers consumed in the Ethernet switch, broadband gateway router, edge router, core router, and WDM link respectively; and $\widehat{\Phi}(\cdot)$ represents the maximum capacity (or bandwidth) of the corresponding network equipment. The number of core routers are doubled to reflect the hardware redundancy of the core network, while the number of WDM links are halved to reflect the core hops between co-located equipment. The overall factor of four further covers extra power consumption under the redundancy policy (factor of 2) and high power expenditure at low network utilization (factor of 2). Note that we removed the factor of 1.5 for cooling and other overheads from the original study.

Similarly, by assuming two hops (over one switch, one edge router, and one gateway router) for accessing a server within a data center [1], [30], the energy consumption of user traffic with respect to both the Cloudlet and the Cloud can be modeled as

$$\begin{aligned} E^{cloud}(n_{net}) &= E^{cloudlet}(n_{net}) \\ &= 4 \cdot \left(\frac{P(r_{switch})}{\widehat{\Phi}(r_{switch})} + \frac{P(r_{edge})}{\widehat{\Phi}(r_{edge})} + \frac{P(r_{gateway})}{\widehat{\Phi}(r_{gateway})} \right) \cdot D(n_{net}), \end{aligned} \tag{31}$$

where $P(r_{gateway})$ and $\widehat{\Phi}(r_{gateway})$ respectively indicate the power and the maximum capacity of the gateway router.

4.5.5 Storage Energy Consumption Model

Given a storage-intensive task n_{disk}, in addition to the data input/output analysis [102] in alignment with Equation (11), the major concern is about accessing data stored in hard disk arrays through content servers [1]. Naturally, the energy consumption of n_{disk} can be split into two parts occurred in the disk arrays (i.e., $E^{array}(n_{disk})$) and content servers (i.e., $E^{server}(n_{disk})$) respectively

$$E(n_{disk}) = E^{array}(n_{disk}) + E^{server}(n_{disk}). \tag{32}$$

Suppose the data $D(n_{disk})$ involved in, or to be accessed by, the task n_{disk} are pre-stored in the disk array r_{array} (for the case of writing, we assume that the same size of storage area has been pre-booked in the disk array). Then, the energy for storing the data during the lifecycle $T(n_{disk})$ of the task can be calculated through

$$E^{array}(n_{disk}) = 2 \cdot D(n_{disk}) \cdot \frac{P(r_{array})}{\widehat{D}(r_{array})} \cdot T(n_{disk}), \tag{33}$$

where $P(r_{array})$ indicates the power of the hard disk array; $\widehat{D}(r_{array})$ stands for the disk content capacity; and the initial factor of 2 accounts for the redundancy policy in storage. As before, we removed the factor of 1.5 that reflects cooling and extra overheads for the power of the hard disk array.

For the purpose of conciseness, we define each task n_{disk} to include only a one-shot access to the data $D(n_{disk})$, and multiple data accesses can be viewed as multiple tasks. Then, the data accessing energy consumed in a content server r_{server} has been modeled by focusing either on the accessing time [64] or on the data size [1]

$$\begin{aligned} E^{server}(n_{disk}) &= P(r_{server,disk}) \cdot T(n_{disk}) \\ &= D(n_{disk}) \cdot \frac{P(r_{server,disk})}{\widehat{\Phi}(r_{server})}, \end{aligned} \tag{34}$$

where $P(r_{server,disk})$ refers to the power consumed in the storage component of r_{server}, and $\widehat{\Phi}(r_{server})$ represents the maximum data throughput over r_{server}. In particular, the factor of extra power requirement for other overheads can also be added to Equation (34) [30].

If allowing multiple clients to access data simultaneously within the same task n_{disk}, the energy consumption located at r_{server} between time t_1 and t_2 was given in [55], [56] without emphasizing the storage component

$$\Delta E^{server}(n_{disk}) = \int_{t_1}^{t_2} \alpha \cdot (P_{idle}(r_{server}) + \beta_t \cdot \Phi(r_{server}, t)) \cdot dt, \tag{35}$$

where α depends on the content server type, $\beta_t \geq 1$ is proportional to the number of clients at time t, and $\Phi(r_{server}, t)$ refers to the data throughput over r_{server} at time t.

4.5.6 Summary

Given the identified 30+ models, it is evident that there is no one-size-fits-all approach to modeling energy consumption of Cloud applications. Various energy consumption models are applied to different situations when emphasizing and combining different factors. By deconstructing and analyzing the existing models, however, we see a regular pattern in the modeling efforts, i.e., on the power characteristics of the resources together with the way resources are utilized by application workloads. This regular pattern confirms the statement that a Cloud application's energy consumption involves a mutual effect between its workload and environmental factors.

Furthermore, by distinguishing between different power consumption components, we see three viewpoints about the energy consumption of Cloud applications, and we name them as Effective, Active, and Incremental energy consumptions.

Effective Energy Consumption, i.e., $E(A) = E_{active}(R(A)) + E_{idle}(R(A))$, includes both the active and the idle power consumed in the environmental resources of a Cloud application. In particular, the idle power consumption is included for two reasons. First, the idle Cloud resources would have to keep a standby state and wait for new jobs, so that they can be rented again at any time [97]. Second, the idle power consumption will still be meaningful and effective if it is used for maintaining the application accessibility and/or the data availability [3].

Active Energy Consumption, i.e., $E(A) = E_{active}(R(A))$, includes only the active power consumed in the environmental resources of a Cloud application. Although the idle power consumption should not be excluded in practice as mentioned above, focusing on the active power consumption would be useful for investigating the energy consumption incurred by dynamic application activities.

Incremental Energy Consumption, i.e., $E(A) = E_{active}(R(A)) + E_{idle}(R(A)) - P_{idle}(R(A)) \cdot T(A)$, is related to the increased power arising from the idle power consumed in the environmental resources of a Cloud application. In other words, the arising power consumption is the top-up part within active power consumption based on its idle counterpart. Since various IT equipment has widely different dynamic power ranges (e.g., network devices operating at the utilization less than 50 percent may still incur nearly the maximum power consumption) [1], [3], emphasizing the incremental energy consumption can reduce possible investigation bias by excluding the background noises [70], [90].

5 Trade-Off Debates

As mentioned in Sections 4.4 and 4.3, we try to isolate the influences of individual factors on the energy consumption, to avoid the combinatorial explosion of the factorial discussions. However, it is noteworthy that the energy expense of a Cloud application is inevitably affected as a result of combining multiple factors, as demonstrated in the mathematical models (cf. Section 4.5). Although studying various combinational factors' effects on the energy consumption is out of the scope of this survey, we particularly highlight a set of trade-off debates that would be worth further investigations, and we believe that model-based simulations would be the key to investigating those concerns raised by these debates.

- *Resource Allocation Level:* To improve the energy efficiency for a Cloud application, there is evidence advocating both less than and more than enough resource allocations. By provisioning "under-the-just-enough" servers, the authors [105] showed that a data-intensive Cloud application can save up to 24 percent in energy consumption with a loss of around 6 percent only in execution time. However, in general cases, Cloud applications are supposed to achieve greater energy efficiency by utilizing more processors, in order to finish more quickly and free the processors sooner. In other words, the energy saving for a Cloud application can be realized by returning its environmental resources to the idle state earlier [32].
- *Degree of Application Parallelism:* This debate can be viewed as a counterpart of the above one from the application's perspective. By tailoring the resource allocations to the degree of parallelism [90], the overall energy consumption can decrease significantly with improved processing concurrency in a Cloud application [91]. This is because the increased parallelism would have more chances to reduce the processing time and overwhelm the influence of the resource increase [89]. Nevertheless, considering the theoretical limit of energy saving of parallel executions [32], it is impossible to infinitely enhance the energy efficiency of a Cloud application by increasing its parallelism degree, not to mention that the increased overhead of process scheduling would meanwhile cause more energy consumption [61], [68].
- *Downscaling CPU Frequency:* In addition to the conflicting opinions on the effectiveness of adjusting CPU frequencies (cf. Section 4.3.6), there is also a debate on energy saving by downscaling the CPU frequency. Considering the cubic relationship between a CPU's power and its clock frequency (cf. Equation (13)), in theory, three quarters of the energy can be saved by halving the processor's clock speed, although the execution time doubles [47]. In practice, unfortunately, blindly downscaling CPU frequency often increases energy consumption [10], and computation-intensive applications would particularly be less energy efficient when operating processors at lower frequencies [91]. Such a debate is still driven by the aforementioned "race to idle", depending on if reducing power consumption can bring overwhelming energy benefits.
- *Workload Offloading:* In mobile Cloud computing, offloading local workloads to external resources have widely been considered effective to shorten

applications' execution time and extend mobile devices' battery life, because powerful remote servers can generally offer a significant speedup for mobile applications [49], [79]. However, simply offloading workloads has been proven not always to be energy efficient [35], unless the workload is characterized by a relatively small communication-computation ratio [47]. Correspondingly, the communication-computation ratio has frequently been employed as a trade-off indicator to help determine the right circumstances of workload offloading [11], [37], [96].

6 Conclusions and Future Work

The energy consumption of Cloud computing is predicted to keep growing and even quadruple the current annual consumption by 2020 [105]. Thus, efficient use of computing power and energy consumption management have become crucial topics for engineering Cloud applications. With modeling as a prevalent approach to addressing energy consumption, a substantially large number of models with a high variety has emerged. This drives us to use SLR as a rigorous surveying approach to study the existing modeling efforts as evidence to build up a knowledge foundation for investigating Cloud applications' energy consumption.

In particular, by deconstructing Cloud computing scenarios, we find that the controllable environmental components (especially client devices) and the application execution elements related to task processing and data communication have attracted most of the research attention as well as the modeling efforts. By identifying energy-related factors, this survey confirms computation and communication to be the existing researchers' major concerns about energy consumption of Cloud applications. Correspondingly, Task Size and Data Size have been considered to be the main workload factors, which would largely interact with CPU Clock Frequency and Network Bandwidth (and Access Point Technology used in the client devices) as main environmental factors. On the contrary, the energy consumption of data storage has attracted little attention, and few studies have intensively investigated and modeled the energy for Cloud applications' memory footprints. Such a finding indicates crucial research gaps that require further research efforts in the future.

In fact, storage policies in different cloud environments, which partly relates to the application's nature, may result in a considerably high persistence of the application's data in the Cloud storage, and in turn gives rise to energy consumption for keeping the data. Not to mentions that the degree of data distribution (for protection purposes) can also negatively affect the energy consumption of data storage. Meanwhile, given the increasing trend of in-memory Cloud computing (e.g., Apache Spark[3]), memory has become a significant contributor to the power consumption in Cloud infrastructures [106].

More importantly, our work has advocated divide-and-conquer to be a principle approach to studying energy consumption in the Cloud computing domain. On one hand, decomposing an energy consumption scenario can help clarify the atomic energy concerns and mitigate the complexity in the corresponding problem. On the other hand, gradually recomposing major energy concerns can facilitate iterative and incremental development of energy consumption models, in order to address the complicated trade-offs and even debates with respect to energy efficiency. Naturally, we will unfold our future work along two directions. The first direction is to gradually expand the knowledge artefact (including both factors and models) established in this survey. As mentioned earlier, various aspects and features of Cloud applications might all have impacts on their energy consumption during their whole lifecycle. We will gradually take them (like design and programming) into account from the perspective of software engineering. The second direction is to implement model-driven simulations to reveal further knowledge about the combinational factorial effects on Cloud applications' energy consumption.

3. http://spark.apache.org/

Acknowledgments

This work is funded in part by the Swedish Research Council (VR) under contract number C0590801 for the project Cloud Control, as well as by the strategic research program eSSENCE. Maria Kihl is a member of the Lund Center for Control of Complex Engineering Systems (LCCC) funded by the Swedish Research Council (VR), the Excellence Center Linköping - Lund in Information Technology (ELLIIT), and the Wallenberg Autonomous Systems and Software Program (WASP).

References

[1] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[2] E. Feller, L. Ramakrishnana, and C. Morin, "Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study," *J. Parallel Distrib. Comput.*, vol. 79–80, pp. 80–89, May 2015.

[3] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[4] 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.

[5] J. Shuja, et al., "Survey of techniques and architectures for designing energy-efficient data centers," *IEEE Syst. J.*, vol. 10, no. 2, pp. 507–519, Jun. 2016.

[6] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 732–794, Jan./Mar. 2016.

[7] Reference.com, "What is a model in science?" Aug. 2016. [Online]. Available: https://www.reference.com/science/model-science-727cde390380e207#

[8] S. J. Mellor, A. N. Clark, and T. Futagami, "Model-driven development – guest editor's introduction," *IEEE Softw.*, vol. 20, no. 5, pp. 14–18, Sep./Oct. 2003.

[9] B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," School Comput. Sci. Math., Keele Univ. Durham Univ., Durham, U.K., Tech. Rep. EBSE 2007-001, 2007.

[10] K. Liu, G. Pinto, and Y. D. Liu, "Data-oriented characterization of application-level energy optimization," in *Fundamental Approaches to Software Engineering*. Berlin, Germany: Springer, 2015, pp. 316–331.

[11] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in Cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, 22 Jun. 2010, pp. 1–7.

[12] J.-Y. L. Boudec, *Performance Evaluation of Computer and Communication Systems*. Lausanne, Switzerland: EPFL Press, Feb. 2011.

[13] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "A factor framework for experimental design for performance evaluation of commercial cloud services," in *Proc. 4th IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 3–6 Dec. 2012, pp. 169–176.

[14] T. Hönig, C. Eibel, R. Kapitza, and W. S. Preikschat, "SEEP: Exploiting symbolic execution for energy-aware programming," *ACM SIGOPS Operating Syst. Rev.*, vol. 45, no. 3, pp. 58–62, Dec. 2011.

[15] N. Siegmund, M. Rosenmüller, and S. Apel, "Automating energy optimization with features," in *Proc. 2nd Int. Workshop Feature-Oriented Softw. Develop.*, 10 Oct. 2010, pp. 2–9.

[16] D. Armstrong, R. Kavanagh, and K. Djemame, "Towards an interoperable energy efficient cloud computing architecture - practice & experience," in *Proc. 4th IEEE Int. Conf. Commun. Workshop*, 6–10 Dec. 2015, pp. 1807–1812.

[17] J. Leverich and C. Kozyrakis, "On the energy (in)efficiency of Hadoop clusters," *ACM SIGOPS Operating Syst. Rev.*, vol. 44, no. 1, pp. 61–65, Jan. 2010.

[18] W. Tian, Q. Xiong, and J. Cao, "An online parallel scheduling method with application to energy-efficiency in cloud computing," *J. Supercomputing*, vol. 66, no. 3, pp. 1773–1790, Dec. 2013.

[19] C. Cappiello, et al., "Monitoring and assessing energy consumption and CO_2 emissions in Cloud-based systems," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 13–16 Oct. 2013, pp. 115–120.

[20] J. Song, Y. Cui, M. Li, J. Qiu, and R. Buyya, "Energy-traffic tradeoff cooperative offloading for mobile cloud computing," in *Proc. IEEE 22nd Int. Symp. Quality Service*, 26–27 May 2014, pp. 284–289.

[21] P. Balakrishnan and C.-K. Tham, "Energy-efficient mapping and scheduling of task interaction graphs for code offloading in mobile cloud computing," in *Proc. IEEE/ACM 6th Int. Conf. Utility Cloud Comput.*, 9–12 Dec. 2013, pp. 34–41.

[22] I. M. Murwantara and B. Bordbar, "A simplified method of measurement of energy consumption in cloud and virtualized environment," in *Proc. IEEE 4th Int. Conf. Big Data Cloud Comput.*, 3–5 Dec. 2014, pp. 654–661.

[23] J. Singh, K. Naik, and V. Mahinthan, "Impact of developer choices on energy consumption of software on servers," in *Proc. Int. Conf. Soft Comput. Softw. Eng.*, 5–6 Mar. 2015, pp. 385–394.

[24] Y. Xiao, et al., "Modeling energy consumption of data transmission over Wi-Fi," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1760–1773, Aug. 2014.

[25] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 625–637, Jun. 2011.

[26] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[27] M. Gribaudo, T. T. N. Ho, B. Pernici, and G. Serazzi, "Analysis of the influence of application deployment on energy consumption," in *Energy Efficient Data Centers*. Cham, Switzerland: Springer, 2015, pp. 87–101.

[28] H. Wu, D. Huang, and S. Bouzefrane, "Making offloading decisions resistant to network unavailability for mobile cloud collaboration," in *Proc. 9th IEEE Int. Conf. Collaborative Comput.: Netw. Appl. Worksharing*, 20–23 Oct. 2013, pp. 168–177.

[29] Z. Li, H. Zhang, L. O'Brien, R. Cai, and S. Flint, "On evaluating commercial cloud services: A systematic review," *J. Syst. Softw.*, vol. 86, no. 9, pp. 2371–2393, Sep. 2013.

[30] M. Altamimi and K. Naik, "The concept of a mobile cloud computing to reduce energy cost of smartphones and ICT systems," in *Information and Communication on Technology for the Fight Against Global Warming*. Berlin, Germany: Springer, 2011, pp. 79–86.

[31] L. Ren and L. Zhang, "An efficient it energy-saving approach based on cloud computing for networked green manufacturing," *Adv. Mater. Res.*, vol. 139–141, pp. 1374–1377, Oct. 2010.

[32] D. Bonner and A. S. Namin, "An energy model for applications running on multicore systems," in *Proc. 2nd Int. Conf. Cloud Green Comput.*, 1–3 Nov. 2012, pp. 1–8.

[33] M. Ali, "Green cloud on the horizon," in *Cloud Computing*. Berlin, Germany: Springer, 2009, pp. 451–459.

[34] L. Liu, et al., "GreenCloud: A new architecture for green data center," in *Proc. 6th Int. Conf. Ind. Session Autonomic Comput. Commun. Ind. Session*, 15–19 Jun. 2009, pp. 29–38.

[35] A. Vishwanath, F. Jalali, K. Hinton, T. Alpcan, R. W. A. Ayre, and R. S. Tucker, "Energy consumption comparison of interactive cloud-based and local applications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 616–626, Apr. 2015.

[36] H. Yuan, C.-C. J. Kuo, and I. Ahmad, "Energy efficiency in data centers and cloud-based multimedia services: An overview and future directions," in *Proc. 2010 IGCC*, 15–18 Aug. 2010, pp. 1–8.

[37] A. Mtibaa, K. A. Harras, and A. Fahim, "Towards computational offloading in mobile device clouds," in *Proc. 5th IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 2–5 Dec. 2013, pp. 331–338.

[38] X. Ma, C. Lin, X. Xiang, and C. Chen, "Game-theoretic analysis of computation offloading for cloudlet-based mobile cloud computing," in *Proc. 18th ACM Int. Conf. Model. Anal. Simul. Wireless Mobile Syst.*, 2–6 Nov. 2015, pp. 271–278.

[39] A. Ravi and S. K. Peddoju, "Handoff strategy for improving energy efficiency and cloud service availability for mobile devices," *Wireless Pers. Commun.*, vol. 81, no. 1, pp. 101–132, Mar. 2015.

[40] Z. Sheng, C. Mahapatra, V. C. Leung, M. Chen, and P. K. Sahu, "Energy efficient cooperative computing in mobile wireless sensor networks," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2015.2458272.

[41] K. Hinton, J. Baliga, M. Feng, R. Ayre, and R. S. Tucker, "Power consumption and energy efficiency in the Internet," *IEEE Netw.*, vol. 25, no. 2, pp. 6–12, Mar./Apr. 2011.

[42] J. Baliga, K. Hinton, and R. S. Tucker, "Energy consumption of the internet," in *Proc. Australian Conf. Opt. Fibre Technol.*, 24–27 Jun. 2007, pp. 1–3.

[43] Z. Sheng, X. Hu, P. TalebiFard, V. C. Leung, R. Chen, and Y. Zhou, "Sensor cloud computing for vehicular applications: Fom analysis to practical implementation," in *Proc. 4th ACM Int. Symp. Develop. Anal. Intell. Veh. Netw. Appl.*, 21–26 Sep. 2014, pp. 53–59.

[44] A. Mtibaa, M. A. Snobery, A. Carelliy, R. Beraldiy, and H. Alnuweiri, "Collaborative mobile-to-mobile computation offloading," in *Proc. 10th IEEE Int. Conf. Collaborative Comput.: Netw. Appl. Worksharing*, 22–25 Oct. 2014, pp. 460–465.

[45] M. Akram and A. ElNahas, "Energy-aware offloading technique for mobile cloud computing," in *Proc. 3rd Int. Conf. Future Internet Things Cloud*, 24–26 Aug. 2015, pp. 349–356.

[46] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3317–3329, Dec. 2015.

[47] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Comput.*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[48] S. A. Saab, F. Saab, A. Kayssi, A. Chehab, and I. H. Elhajj, "Partial mobile application offloading to the cloud for energy-efficiency with security measures," *Sustainable Comput. Inf. Syst.*, vol. 8, pp. 38–46, Dec. 2015.

[49] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, 10–13 Apr. 2011, pp. 301–314.

[50] J. Kim, "Design and evaluation of mobile applications with full and partial offloading," in *Advances in Grid and Pervasive Computing*. Berlin, Germany: Springer, 2012, pp. 172–182.

[51] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[52] W. Gao, Y. Li, H. Lu, T. Wang, and C. Liu, "On exploiting dynamic execution patterns for workload offloading in mobile cloud applications," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, 21–24 Oct. 2014, pp. 1–12.

[53] C.-H. Lin, P.-C. Hsiu, and C.-K. Hsieh, "Dynamic backlight scaling optimization: A cloud-based energy-saving service for mobile streaming applications," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 335–348, Feb. 2014.

[54] M. Segata, B. Bloessl, C. Sommer, and F. Dressler, "Towards energy efficient smart phone applications: Energy models for offloading tasks into the Cloud," in *Proc. IEEE Int. Conf. Commun.*, 10–14 Jun. 2014, pp. 2394–2399.

[55] T. Enokido, K. Suzuki, A. Aikebaier, and M. Takizawa, "Algorithms for reducing the total power consumption in data communication-based applications," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, 20–23 Apr. 2010, pp. 142–149.

[56] T. Enokido, K. Suzuki, A. Aikebaier, and M. Takizawa, "Laxity based algorithm for reducing power consumption in distributed systems," in *Proc. 4th Int. Conf. Complex Intell. Softw. Intensive Syst.*, 15–18 Feb. 2010, pp. 321–328.

[57] X. Xiang, C. Lin, and X. Chen, "EcoPlan: Energy-efficient downlink and uplink data transmission in mobile cloud computing," *Wireless Netw.*, vol. 21, no. 2, pp. 453–466, Feb. 2015.

[58] 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. Syst. Integr.*, 20–22 Dec. 2011, pp. 1159–1164.

[59] F. Chen, J. Grundy, J.-G. Schneider, Y. Yang, and Q. He, "Automating performance and energy consumption analysis for cloud applications," in *Proc. 11th IEEE World Congr. Services*, 27 Jun.-2 Jul. 2015, pp. 63–70.

[60] L. Pu, J. Xu, X. Jin, and J. Zhang, "SmartVirtCloud: Virtual cloud assisted application offloading execution at mobile devices' discretion," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 7–10 Apr. 2013, pp. 4398–4403.

[61] F. Chen, J.-G. Schneider, Y. Yang, J. Grundy, and Q. He, "An energy consumption model and analysis tool for cloud computing environments," in *Proc. 1st Int. Workshop Green Sustainable Softw.*, 3 Jun. 2012, pp. 45–50.

[62] C. Luo, L. T. Yang, P. Li, X. Xie, and H.-C. Chao, "A holistic energy optimization framework for cloud-assisted mobile computing," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 118–123, Jun. 2015.

[63] 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.

[64] S. Izadpanah, K. Pawlikowski, F. Davoli, and D. McNickle, "Evaluation of energy consumption and data access time in data fetching in grid-based data-intensive applications," in *Proc. 22nd ITC Specialist Seminar Energy Efficient Green Netw.*, 20–22 Nov. 2013, pp. 37–42.

[65] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: A packet-level simulator of energy-aware cloud computing data centers," *J. Supercomputing*, vol. 62, no. 3, pp. 1263–1283, Dec. 2012.

[66] G. Folino and F. S. Pisani, "Modeling the offloading of different types of mobile applications by using evolutionary algorithms," in *Applications of Evolutionary Computation*. Berlin, Germany: Springer, 2014, pp. 86–97.

[67] C.-h. Hsu, C. chin Lin, and T. sheng Hsu, "Energy-conscious cloud computing adopting DVFS and state-switching for workflow applications," in *Proc. Int. Conf. Cloud Comput. Big Data*, 16–19 Dec. 2013, pp. 1–8.

[68] 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 Int. Conf. Performance Eng.*, 21–24 Apr. 2013, pp. 295–306.

[69] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On the conceptualization of performance evaluation of IaaS services," *IEEE Trans. Services Comput.*, vol. 7, no. 4, pp. 628–641, Oct.-Dec. 2014.

[70] F. Jalali, et al., "Energy consumption of photo sharing in online social networks," in *Proc. 14th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, 26–29 May 2014, pp. 604–611.

[71] S. A. Saab, A. Chehab, and A. Kayssi, "Energy efficiency in mobile cloud computing: Total offloading selectively works. does selective offloading totally work?" in *Proc. 4th Annu. Int. Conf. Energy Aware Comput. Syst. Appl.*, 16–18 Dec. 2013, pp. 164–168.

[72] K. Fekete, K. Csorba, B. Forstner, M. Fehér, and T. Vajk, "Energy-efficient computation offloading model for mobile phone environment," in *Proc. IEEE 1st Int. Conf. Cloud Netw.*, 28–30 Nov. 2012, pp. 95–99.

[73] Q. Xia, W. Liang, Z. Xu, and B. Zhou, "Online algorithms for location-aware task offloading in two-tiered mobile cloud environments," in *Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput.*, 8–11 Dec. 2014, pp. 109–116.

[74] P. Yuan, Y. Guo, and X. Chen, "Uniport: A uniform programming support framework for mobile cloud computing," in *Proc. 3rd IEEE Int. Conf. Mobile Cloud Comput. Services Eng.*, 30 Mar.–3 Apr. 2015, pp. 71–80.

[75] H. Wu and K. Wolter, "Software aging in mobile devices: Partial computation offloading as a solution," in *Proc. Int. Symp. Softw. Rel. Eng. Workshops*, 2–5 Nov. 2015, pp. 125–131.

[76] F. Xia, F. Ding, J. Li, X. Kong, L. T. Yang, and J. Ma, "Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing," *Inf. Syst. Frontiers*, vol. 16, no. 1, pp. 95–111, Mar. 2014.

[77] H. M. Fard, R. Prodan, J. J. D. Barrionuevo, and T. Fahringer, "A multi-objective approach for workflow scheduling in heterogeneous environments," in *Proc. 12th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, 13–16 May 2012, pp. 300–309.

[78] D. Mazza, D. Tarchi, and G. E. Corazza, "A user-satisfaction based offloading technique for smart city applications," in *Proc. IEEE Global Commun. Conf.*, 8–12 Dec. 2014, pp. 2783–2788.

[79] T. Nabi, P. Mittal, P. Azimi, D. Dig, and E. Tilevich, "Assessing the benefits of computational offloading in mobile-cloud applications," in *Proc. 3rd Int. Workshop Mobile Develop. Lifecycle*, 26 Oct. 2015, pp. 17–24.

[80] F. Wu, Q. Wu, Y. Tan, and W. Wang, "Unified multi-constraint and multi-objective workflow scheduling for cloud system," in *Algorithms and Architectures for Parallel Processing*. Cham, Switzerland: Springer, 2015, vol. 9529, pp. 635–650.

[81] R. G. Babukarthik, R. Raju, and P. Dhavachelvan, "Energy-aware scheduling using hybrid algorithm for cloud computing," in *Proc. 3rd Int. Conf. Comput. Commun. Netw. Technol.*, 26–28 Jul. 2012, pp. 1–6.

[82] R. Ge, X. Feng, S. Subramanya, and X. he Sun, "Characterizing energy efficiency of i/o intensive parallel applications on power-aware clusters," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops PhD Forum*, 19–23 Apr. 2010, pp. 1–8.

[83] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of cloud resources for real-time services," in *Proc. 7th Int. Workshop Middleware Grids Clouds e-Sci.*, 1 Dec. 2009, pp. 1–6.

[84] Z. Ou, B. Pang, Y. Deng, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Energy- and cost-efficiency analysis of arm-based clusters," in *Proc. 12th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, 13–16 May 2012, pp. 115–123.

[85] T. Thanavanich and P. Uthayopas, "Efficient energy aware task scheduling for parallel workflow tasks on hybrids cloud environment," in *Proc. 17th Int. Comput. Sci. Eng. Conf.*, 4–6 Sep. 2013, pp. 37–42.

[86] H. S. Abdelsalam, K. Maly, R. Mukkamala, M. Zubair, and D. Kaminsky, "Analysis of energy efficiency in clouds," in *Proc. Computation World: Future Comput. Service Comput. Cognitive Adaptive Content Patterns*, 15–20 Nov. 2009, pp. 416–421.

[87] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron, "PowerPack: Energy profiling and analysis of high-performance systems and applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 658–671, May 2010.

[88] S. Ibrahim, D. Moise, H.-E. Chihoub, A. Carpen-Amarie, L. Bougé, and G. Antoniu, "Towards efficient power management in MapReduce: Investigation of cpu-frequencies scaling on power efficiency in Hadoop," in *Adaptive Resource Management and Scheduling for Cloud Computing*. Cham, Switzerland: Springer, 2014, pp. 147–164.

[89] L. Wang, G. von Laszewski, J. Dayal, and F. Wang, "Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS," in *Proc. 10th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, 17–20 May 2010, pp. 368–377.

[90] T. Wirtz and R. Ge, "Improving MapReduce energy efficiency for computation intensive workloads," in *Proc. 2nd Int. Green Comput. Conf. Workshops*, 25–28 Jul. 2011, pp. 1–8.

[91] Z. Zhang and S. Fu, "Characterizing power and energy usage in cloud computing systems," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci.*, 29 Nov.-1 Dec. 2011, pp. 146–153.

[92] K. H. Kim, R. Buyya, and J. Kim, "Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters," in *Proc. 7th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, 14–17 May 2007, pp. 541–548.

[93] Q. Huang, S. Su, J. Li, P. Xu, K. Shuang, and X. Huang, "Enhanced energy-efficient scheduling for parallel applications in cloud," in *Proc. 12th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, 13–16 May 2012, pp. 781–786.

[94] T. Enokido and M. Takizawa, "Power consumption and computation models of virtual machines to perform computation type application processes," in *Proc. 9th Int. Conf. Complex Intell. Softw. Intensive Syst.*, 8–10 Jul. 2015, pp. 126–133.

[95] A. Bergen, R. Desmarais, S. Ganti, and U. Stege, "Towards software-adaptive green computing based on server power consumption," in *Proc. 3rd Int. Workshop Green Sustainable Softw.*, 1 Jun. 2014, pp. 9–16.

[96] V. Namboodiri and T. Ghose, "To cloud or not to cloud: A mobile device perspective on energy consumption of applications," in *Proc. 13th IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, 25–28 Jun. 2012, pp. 1–9.

[97] W. Zheng and S. Huang, "Deadline constrained energy-efficient scheduling for workflows in clouds," in *Proc. 2nd Int. Conf. Adv. Cloud Big Data*, 20–22 Nov. 2014, pp. 69–76.

[98] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services*, 25–29 Jun. 2012, pp. 225–238.

[99] W. Liu, J. Cao, X. Qiu, and J. Li, "Improving performance of mobile interactive data-streaming applications with multiple cloudlets," in *Proc. 6th IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 15–18 Dec. 2014, pp. 46–53.

[100] U. Wajid, C. A. Marín, and A. Karageorgos, "Optimizing energy efficiency in the cloud using service composition and runtime adaptation techniques," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 13–16 Oct. 2013, pp. 115–120.
[101] J.-C. Lin, F.-Y. Leu, and Y.-P. Chen, "Analyzing job completion reliability and job energy consumption for a heterogeneous MapReduce cluster under different intermediate-data replication policies," *J. Supercomputing*, vol. 71, no. 5, pp. 1657–1677, May 2015.
[102] P. Xiao, Z.-G. Hu, and Y.-P. Zhang, "An energy-aware heuristic scheduling for data-intensive workflows in virtualized datacenters," *J. Comput. Sci. Technol.*, vol. 28, no. 6, pp. 948–961, Nov. 2013.
[103] L. Zhang, K. Li, and K. Li, "Bi-objective optimization genetic algorithm of the energy consumption and reliability for workflow applications in heterogeneous computing systems," in *Algorithms and Architectures for Parallel Processing*. Cham, Switzerland: Springer, 2015, pp. 651–664.
[104] P. Chaparro, J. González, and A. González, "Thermal-effective clustered microarchitectures," in *Proc. 1st Int. Symp. Theoretical Aspects Comput. Softw.*, Jun. 2004, pp. 1–9.
[105] D. Guyon, A.-C. Orgerie, and C. Morin, "Energy-efficient user-oriented cloud elasticity for data-driven applications," in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, 11–13 Dec. 2015, pp. 376–383.
[106] M. Chen, X. Wang, and X. Li, "Coordinating processor and main memory for efficientserver power control," in *Proc. 25th Int. Conf. Supercomputing*, 31 May–4 Jun. 2011, pp. 130–140.

Zheng Li received the BEng degree from Zhengzhou University and the MScEng degree from the Beijing University of Chemical Technology. He received the PhD degree and ME by research degree from the Australian National University (ANU) and the University of New South Wales (UNSW), respectively. He is a postdoctoral researcher with Lund University, Sweden. During the same time, he was a graduate researcher with the Software Systems Research Group (SSRG) at National ICT Australia (NICTA). Before studying abroad, he had around four-year industrial experience in China after receiving his BEng and MScEng degrees. His research interests include cloud computing, performance engineering, empirical software engineering, software cost/effort estimation, and Web service composition. He is a member of the IEEE.

Selome Tesfatsion received the MS degree in computing science from Umeå University, in 2013. She is currently working toward the PhD degree in the Department of Computing Science, Umeå University. Her main research interests include energy management, cloud computing, virtualization, performance modeling, and data center management. She is a student member of the IEEE.

Saeed Bastani received the BSc degree in software engineering from the Isfahan University of Technology, Isfahan, Iran, in 1998, the MSc degree in artificial intelligence and robotics from the Iran University of Science and Technology, Tehran, Iran, in 2002, and the PhD degree in computer science from the University of Sydney, Australia, in 2002. Prior to his PhD studies, he worked for seven years as a researcher and developer in the telecommunication industries. After completing his PhD studies, he worked for six months as a postdoctoral researcher, focused on vehicular communication networks, at the University of New South Wales, Australia. Since March 2014, he has been a researcher with Lund University in Sweden, where he has been involved in European projects in different roles, from research to leadership. His current research interests span both wired and wireless communication networks, including energy-efficient networking, reliability of wireless communications, and efficient distribution of massive content in the Internet. He is a member of the IEEE.

Ahmed Ali-Eldin received the PhD degree from Umeå University, Sweden, in 2015. He is a postdoctoral researcher with Umeå University and UMass, Amherst. His research interests lie in the intersection of computer systems and performance modeling. He is a member of the IEEE.

Erik Elmroth is a professor of computing science with Umeå University. He has been head and deputy head of the Department of Computing Science for 10 years. He is leading the Umeå University research on distributed systems, focusing on theory, algorithms, and systems for the autonomous management of ICT resources, spanning from individual servers to large scale cloud datacenters, federated clouds, highly distributed edge clouds, and software-defined infrastructures. He received the Nordea Scientific Award 2011. Prehistoric highlights include being co-winner of the SIAM Linear Algebra Prize 2000, for the most outstanding linear algebra publication worldwide (in any journal) during the preceding 3-year period. He has until recently been chairman of the Board of the Swedish National Infrastructure for Computing (SNIC). Previously, he served as chairman of the Swedish Research Council's (VR's) expert group on e-science infrastructures and as member of VR's Council for Research Infrastructures (RFI). He is a member of the IEEE.

Maria Kihl is a professor of internetworked systems in the Department of Electrical and Information Technology, Lund University, Sweden. Her main research interests include the fields of performance modeling and analysis of distributed applications and telecommunication systems. Currently, her projects mainly concern optimization and control of cloud infrastructures and streaming Internet applications. She is a member of the IEEE.

Rajiv Ranjan received the PhD degree from the Department of Computer Science and Software Engineering, University of Melbourne, in 2009. He is a reader (equivalent to non-distinguished full professor in the North American System) in computing science with Newcastle University, United Kingdom. Before moving to Newcastle University, he was Julius fellow (2013-2015), senior research scientist, and project leader in the Digital Productivity and Services Flagship of Commonwealth Scientific and Industrial Research Organization (CSIRO C Australian Government's Premier Research Agency). Prior to that, he was a senior research associate (lecturer level B) in the School of Computer Science and Engineering, University of New South Wales (UNSW). He is a member of the IEEE.