A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View

SUKHPAL SINGH GILL and RAJKUMAR BUYYA, The University of Melbourne, Australia

The cloud-computing paradigm offers on-demand services over the Internet and supports a wide variety of applications. With the recent growth of Internet of Things (IoT)–based applications, the use of cloud services is increasing exponentially. The next generation of cloud computing must be energy efficient and sustainable to fulfill end-user requirements, which are changing dynamically. Presently, cloud providers are facing challenges to ensure the energy efficiency and sustainability of their services. The use of a large number of cloud datacenters increases cost as well as carbon footprints, which further affects the sustainability of cloud services. In this article, we propose a comprehensive taxonomy of sustainable cloud computing. The taxonomy is used to investigate the existing techniques for sustainability that need careful attention and investigation as proposed by several academic and industry groups. The current research on sustainable cloud computing is organized into several categories: application design, sustainability metrics, capacity planning, energy management, virtualization, thermal-aware scheduling, cooling management, renewable energy, and waste heat utilization. The existing techniques have been compared and categorized based on common characteristics and properties. A conceptual model for sustainable cloud computing has been presented along with a discussion on future research directions.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey; C.0 [**General**]: Systems Architectures; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.1 [**Process Management**]: Scheduling; H.3.4 [**Systems and Software**]: Distributed Systems; J.7 [**Distributed Parallel and Cluster Computing**]; K.6.2 [**Management of Computing and Information Systems**]: Installation Management

General Terms: Documentation, Cloud Computing, Methodical Analysis, Conceptual Model, Focus of Study, Research Challenges, Theory, Reference Architecture, Trade-off, Future Directions, Management, Systematic Review, Survey

Additional Key Words and Phrases: Energy efficiency, sustainability, cloud datacenters, quality of service, green computing, holistic management, sustainable cloud computing, application design, energy management, renewable energy, thermal-aware scheduling, virtualization, sustainable cloud datacenters, capacity planning, sustainable metrics, cooling management, and waste heat utilization

ACM Reference format:
Sukhpal Singh Gill and Rajkumar Buyya. 2018. A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View. *ACM Comput. Surv.* 51, 5, Article 104 (December 2018), 33 pages.
https://doi.org/10.1145/3241038

This work is supported by the Melbourne-Chindia Cloud Computing (MC3) Research Network and Australian Research Council (DP160102414).

Authors' addresses: S. S. Gill and R. Buyya, Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Parkville, Australia - 3010; emails: sukhpal.gill@unimelb.edu.au, rbuyya@unimelb.edu.au.

0360-0300/2018/12-ART104 $15.00
https://doi.org/10.1145/3241038

1 INTRODUCTION

Cloud computing offers a flexible and powerful computing environment to provide on-demand, subscription-based online services over the Internet to host applications on a pay-as-you-go basis. The various cloud providers, such as Microsoft, Google, and Amazon, make extensive use of Cloud Data Centers (CDCs) to fulfill the requirements (memory, data, compute, or network) of the digital world. To reduce the service delay and maintain the Service Level Agreement (SLA), fault tolerance should be provided through replicating the compute abilities redundantly [1]. To ensure the availability and reliability of services, the components of CDCs—such as network devices, storage devices, and servers—should be run 24/7 [2]. Large amounts of data are created by digital activities such as data streaming, file sharing, searching and social networking websites, e-commerce, and sensor networks. That data can be stored as well as processed efficiently using CDCs [3, 4]. The energy cost is added by creating, processing, and storing each bit of data, which increases carbon footprints that further impacts on the sustainability of cloud services. Due to the large consumption of electricity by CDCs, the research community is addressing the challenge of designing sustainable CDCs [5].

With the continuous growth of Internet of Things (IoT)–based applications, the use of cloud services is increasing exponentially, which further increases the electricity consumption of CDCs by 20% to 25% every year [6]. Existing studies claimed that 78.7 million metric tons of CO_2 are produced by datacenters, which is equal to 2% of global emissions [7]. CDCs in the United States consumed 100 billion kilowatt hours (kWh) in 2015, which is sufficient for powering Washington, DC [11] for a year. The consumption of electricity will reach 150 billion kWh by 2022, that is, increase by 50% [12]. Energy consumption in CDCs can be increased to 8000 terawatt hours (TWh) in 2030 if controlled mechanisms are not identified [122, 81]. Due to underloading and overloading of resources in infrastructure (cooling, computing, storage, networking, etc.), the energy consumption in cloud datacenters is not efficient and the energy is consumed mostly while some of the resources are in idle state, which increases the cost of cloud services [11]. Carbon footprints produced by CDCs are the same as that of the aviation industry [13, 135]. In the current scenario, CDC service providers are finding alternative ways to reduce the carbon footprint of their infrastructure. The prominent cloud providers—such as Google, Amazon, Microsoft, and IBM—have vowed to attain zero production of carbon footprints and they are finding the new ways to make CDCs and cloud-based services eco-friendly [3]. Therefore, CDCs need to provide cloud services with a minimum carbon footprint and minimum heat release in the form of greenhouse gas emissions [71, 136]. The energy consumption of different components of a CDC [3, 71, 95] is shown in Figure 16 of Appendix A. The types of sustainability spheres are shown in Figure 17 of Appendix A. The investment in cloud computing is shown in Figure 18 of Appendix A.

To solve this challenge of energy-efficient cloud services, a large number of researchers proposed resource management policies, algorithms and architectures, but energy efficiency is still a challenge for future researchers. To ensure a high level of sustainability, holistic management of resources can solve new open challenges existing in resource scheduling. There is a need for methods that harness renewable energy to decrease carbon footprints without the use of fossil fuels. Further, cooling expenses can be decreased by developing waste heat utilization and free cooling mechanisms. Location-aware ideal climatic conditions are needed for an efficient implementation of free cooling and renewable energy production techniques [47, 57, 91]. Moreover, waste heat recovery locations are required to be identified for an efficient implantation of waste heat recovery predictions. CDCs can be relocated based on (i) opportunities for waste heat recovery, (ii) accessibility of green computing resources and (iii) proximity of free cooling resources. Cloud providers such as Google, Amazon, IBM, Facebook, and Microsoft are using more green energy resources instead of grid electricity [31, 58].

1.1 Background

The background of sustainable cloud computing is given in Appendix A.

1.2 Related Surveys and Our Work

The comparison of our work with related surveys is presented in Table 1 of Appendix A.1.

1.3 Our Contributions

- A comprehensive taxonomy for sustainable cloud computing is proposed.
- A broad review has been conducted to explore various existing techniques for sustainable cloud computing.
- The current research on sustainable cloud computing is organized into several categories: application design, sustainability metrics, capacity planning, energy management, virtualization, thermal-aware scheduling, cooling management, renewable energy, and waste heat utilization.
- Existing techniques have been compared and categorized based on common characteristics and properties.
- A conceptual model for sustainable cloud computing has been proposed.
- The taxonomy and survey results are used to find the open challenges that have not been fully explored in the research.

1.4 Article Structure

The rest of the article is organized as follows: Section 2 describes the review technique used to find and analyze the available existing research, research questions, and searching criteria. Section 3 presents a proposed comprehensive taxonomy and systematic review of existing techniques for sustainable cloud computing. Based on common characteristics and properties, techniques have been compared and categorized. Section 4 describes the outcomes of this systematic review. Section 5 introduces the open challenges and future research directions along with implications of this research in the area of sustainable cloud computing. Section 6 offers the conceptual model for sustainable cloud computing. Section 7 summarizes this research. ***Note:*** Important information regarding sustainable cloud computing is included in the online Appendix; see the Appendix for the full picture.

2 REVIEW METHODOLOGY

The review technique [45] used in this systematic review is described in Appendix B.

3 SUSTAINABLE CLOUD COMPUTING: A TAXONOMY

The ever-increasing demand for cloud computing services that are deployed across multiple cloud datacenters harnesses significant amount of power, resulting in not only high operational cost but also high carbon emissions [87, 46]. In sustainable cloud computing, the CDCs are powered by renewable energy resources by replacing the conventional fossil fuel-based grid electricity or brown energy to effectively reduce carbon emissions [2]. Employing energy-efficiency mechanisms also makes cloud computing sustainable by reducing carbon footprints to a great extent [144]. Waste heat utilization from heat dissipated through servers and employing mechanisms for free cooling of the servers make the CDCs sustainable [3, 71, 80]. Thus, sustainable cloud computing employs the following elements in making the datacenter sustainable [4]: (i) using renewable energy instead of grid energy generated from fossil fuels, (ii) using the waste heat generated from heat dissipating servers, (iii) using free cooling mechanisms, and (iv) using energy-efficient mechanisms [145]. All of these factors contribute to reducing carbon footprints, operational cost, and energy consumption to make CDCs more sustainable [146, 165, 166]. Figure 20 of Appendix C presents various elements that impact or support sustainable cloud computing (360-Degree View), which have been broken out into nine categories: application design, sustainability metrics, capacity planning, energy management, virtualization, thermal-aware scheduling, cooling management, renewable energy, and waste heat utilization based on the existing literature. Table 6 of Appendix C contains the mapping of aspects of sustainable CDCs to types of sustainability spheres based on Figures 17 and 20.

3.1 Application Design

In sustainable cloud computing, the design of an application plays a vital role and the efficient structure of an application can improve energy efficiency of CDCs. The resource manager and scheduler follow different approaches for application modelling [25, 26]. For example, the scheduling algorithm for the Map Reduce model follows a different approach compared with other models such as workflow, web application, streaming application, and graph processing. To make the infrastructure sustainable and environmentally eco-friendly, there is a need for green ICT-based innovative applications [19, 140, 141, 142]. Effective design of cloud applications contains APIs or services. Applications (e.g., web applications) designed following three-tiered architectures contain user interfaces, application processing, and databases [29, 27, 42]. The functionality of each tier should be independent to run at different providers to improve its performance, simplicity, and reliability [138, 79]. The components of applications should have minimum dependency, that is, they should be loosely coupled. Applications can be ported from one server to another without affecting their execution [43, 44]. At the software level, cloud users can use applications in a flexible manner, which are running on cloud datacenters [78].

Recent technological developments such as the Internet of Things (IoT) and software-defined cloud-based applications are creating new research areas for sustainable cloud computing [1, 56]. The emerging IoT-based applications, such as smart cities and health care services, are increasing, which need appropriate application design model for fast data processing that improves the performance of computing systems [17, 3]. However, these applications are facing high delay and response times because computing systems need to transfer data to the cloud and then from the cloud to an application, which affects the sustainability of cloud computing [2, 24, 137]. Due to a large amount of data processing in the cloud, a computing system does not process at the required speed, which leads to communication failures. Moreover, data security is also a high-priority requirement of sustainable computing to protect critical information from attackers in the case of e-commerce applications [38, 20]. There is a need for reevaluation of existing application models of cloud computing to address research issues such as energy efficiency, sustainability, privacy, security, and reliability. The evolution of application design techniques (see Figure 21) and their comparison along with open research challenges [29, 18, 138, 79, 78, 56, 24, 137, 38, 20, 26, 42, 25, 35, 54, 44] are presented in Table 7 of Appendix C.1.

3.1.1 *Application Design-Based Taxonomy.*

Different types of applications are running in cloud environments, such as computation intensive or data intensive. To improve the performance of cloud computing systems, it is necessary to execute applications in parallel. Based on the requirements of the cloud user, quality of service (QoS) parameters for every application are identified as well as provisions of the resources for execution. The components of the application design taxonomy are (i) QoS parameter, (ii) application model, (iii) workload type, and (iv) type of architecture, as shown in Figure 1. Each of these taxonomy elements are discussed below along with relevant examples. The comparison of existing techniques (discussed in Appendix C.1) based on our application design taxonomy is given in Table 8 of Appendix C.1.

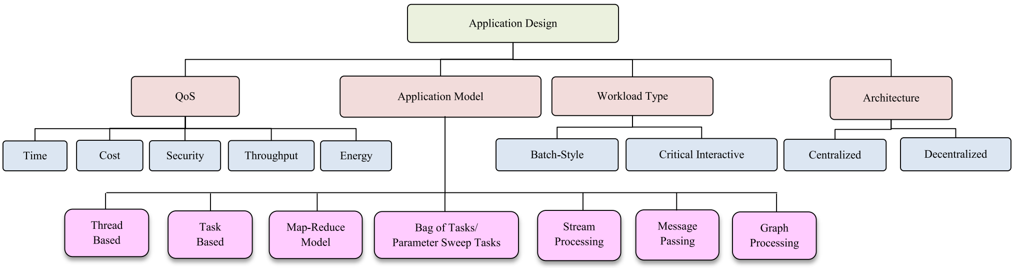

Fig. 1. Taxonomy based on application design.

3.1.1.1 Quality of Service (QoS). Different applications have their different QoS requirements. There are five main types of QoS parameters for sustainable computing as identified from the literature [20, 33, 39, 58, 59, 77], such as execution cost, time, energy consumption, security, and throughput. *Execution cost* is total money that can be spent in 1 hour to execute the application successfully. *Execution time* is the amount of time required to execute an application successfully. *Energy* is the amount of electricity expended by a resource to complete the execution of an application. *Security* is an ability of the computing system to protect the system from malicious attacks. *Throughput* is the ratio of total number of tasks of an application to the amount of time required to execute the tasks. Other QoS requirements of cloud service can be reliability, availability, scalability, and latency.

3.1.1.2 Application Models. The complexity of applications is increasing day by day and the cloud platform can be used to handle user applications. Different types of application models are being developed for a wide range of domains to satisfy the different types of customers for sustainable computing [25, 42-44, 81, 125]. There are seven types of *application models* as identified from the literature [47, 48]: (1) thread-based, (2) task-based, (3) Map-Reduce model, (4) bag-of-tasks or parameter sweep tasks, (5) stream processing, (6) message passing, and (7) graph processing. In the *thread-based* model, one process is divided into multiple threads, which execute concurrently and share resources such as memory, network, and processor to complete execution. In the *task-based* model, a large task is divided into small tasks that are executed in parallel on different non-sharable cloud resources. *Map-reduce tasks* split the input dataset into independent chunks and in a parallel execution, which is used to execute the mapped tasks. Further, the outputs of the maps are sorted and used as an input to the reduce tasks. *Bag-of-tasks* or *parameter sweep tasks* refers to the jobs that are parallel, among which there are no dependencies and are identical in their nature and differ only by the specific parameters used to execute them: for example, video coding and encoding. *Stream processing* is the processing of small-sized data (in kilobytes) generated continuously by thousands of data sources (geospatial services, social networks, mobile or web applications, online gaming, and video streaming), which typically send data records simultaneously. An example of a stream processing model can be a video processing application. The *Message Passing* interface provides a communication functionality between a set of processes, which are mapped to nodes or servers in a language-independent way and encouraged development of portable and scalable large-scale parallel applications. *Graph processing* involves the process of analyzing, storing, and processing graphs to produce effective outputs.

3.1.1.3 Workload Types. For workload management in sustainable cloud computing, there are mainly two types of IT workloads that are considered for sustainable computing: batch style and critical interactive [32, 33]. *Batch-style* workloads are submitted to a job queue and will be executed when resources become available. Multiple batch jobs are often submitted without any deadline

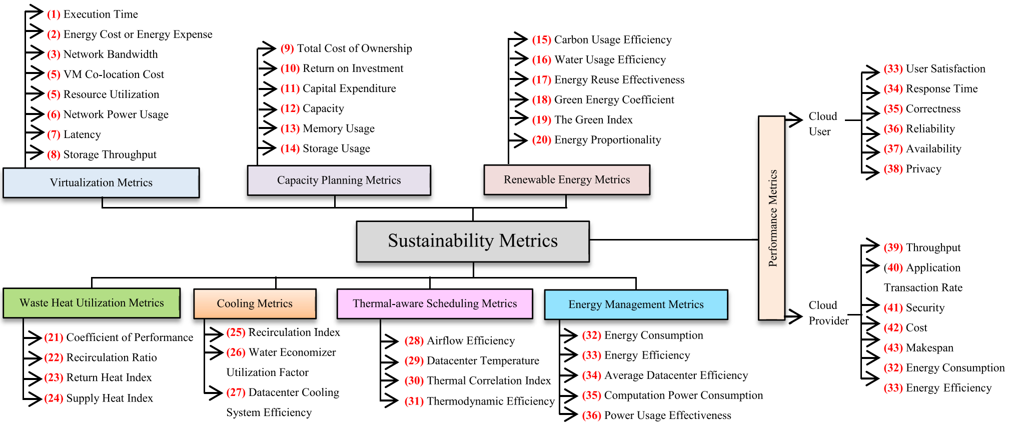

Fig. 2. Taxonomy of metrics for sustainable cloud computing.

constraint together and are executed with maximum resource use. The workloads that need immediate response but whose execution should be completed before their deadline are called *critical interactive* workloads.

3.1.1.4 Architecture. The architecture is an important component of sustainable cloud computing. There are basically two types of architectures: centralized and decentralized [46, 55, 58, 137]. In *centralized* architectures, there is a central controller that manages all the tasks that need to be executed and executes those tasks using scheduled resources. The central controller is responsible for the execution of all tasks. In *decentralized* architectures, resources are allocated independently to execute the tasks without any mutual coordination. Every resource is responsible for its own task execution.

The performance of QoS parameters of different cloud applications is measured using different metrics as discussed in Section 3.2.

3.2 Sustainability Metrics

As use of cloud infrastructure is growing exponentially, it is important to monitor and measure the performance of CDCs regularly. We have identified different types of metrics from the literature [9, 14, 15, 16, 22, 23, 28, 30, 32, 34, 36, 51, 52, 60, 67, 83, 84, 85, 86, 96, 125] and present a taxonomy of metrics for different categories for sustainable cloud computing based on the core operations of CDCs. Figure 2 shows the taxonomy of metrics for application design, capacity planning, energy management, virtualization, thermal-aware scheduling, cooling management, renewable energy, and waste heat utilization. The detailed description of metrics for sustainable cloud computing can be found in [36]. Table 9 (in Appendix C.2) presents the year-wise use of sustainability metrics in different categories of sustainable cloud computing to measure the performance of numerous infrastructure components of CDCs. Table 10 (in Appendix C.2) presents the brief definition of sustainability metrics. Effective capacity planning in the cloud era demands some resource flexibility due to changing application requirements and hosting infrastructure, which is discussed in Section 3.3.

3.3 Capacity Planning

Cloud service providers must initiate effective and organized capacity planning to enable sustainable computing. Capacity planning can be done for power infrastructure, IT devices, and cooling.

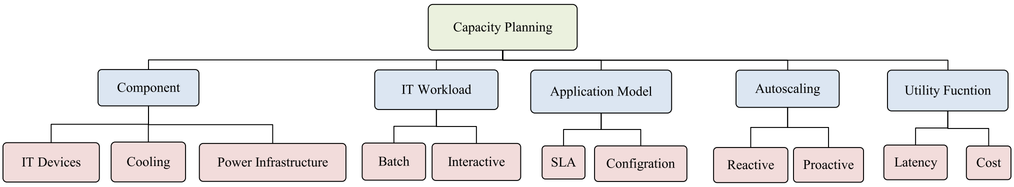

Fig. 3. Taxonomy based on capacity planning.

The capacity of a CDC can be planned effectively by considering the devices of end users, for example, encoding techniques for a video on-demand application [109].

An SLA should be there for important parameters such as backup and recovery, storage, and availability to improve user satisfaction, which attracts more customers in future. There is a need to consider important utilization parameters per application to maximize the use of resources through virtualization by finding the applications, which can be merged. Merging applications improves resource utilization and reduces capacity cost, which makes cloud infrastructure more sustainable. For efficient capacity planning, cloud workloads should be analyzed before execution to finish their execution for deadline-oriented workloads [11, 108]. To manage power infrastructure effectively, virtual machine (VM) migration should be provided for migration of workloads or machines to successfully complete the execution of workloads with minimum use of resources, which improves the energy efficiency of CDCs. Effective capacity planning can truly enable a sustainable cloud environment. The evolution of capacity planning techniques (see Figure 22) and their comparison along with open research challenges [149, 114, 113, 112, 109, 111, 110] can be found in Table 11 of Appendix C.3.

3.3.1 Capacity Planning-Based Taxonomy. Capacity planning is done based on component, IT workload, model, autoscaling, and utility function, as shown in Figure 3. Each of these taxonomy elements are discussed below along with relevant examples. The comparison of existing techniques (discussed in Appendix C.3) based on our capacity planning taxonomy is given in Table 12 of Appendix C.3.

3.3.1.1 Component. Capacity planning is required for every *component* of a CDC, such as IT devices, cooling, and power infrastructures [109, 110, 115]. *IT devices* are an important component, which are required to execute the operations of CDCs. Due to consumption of huge amount of energy, an efficient planning of *cooling* is required to maintain the temperature of CDCs. The planning of the *power infrastructure* is the most important element of a CDC to run it every time, that is, 24 × 7.

3.3.1.2 IT Workload. There are mainly two types of *IT workloads,* which are considered for capacity planning: batch style and critical interactive, as described in Section 3.1.1.3.

3.3.1.3 Application Models. There are two types of design *models* for effective capacity planning: SLA based and configuration based [111, 114]. In the *SLA*-based model, capacity of CDCs is planned based on the QoS requirements of the workloads without SLA violations. The *configuration*-based model focuses on the configuration of the CDC, such as processor, memory, network devices, cooling, and storage, which are required to execute the workloads effectively.

3.3.1.4 Autoscaling. The capacity of a CDC is also planned for *autoscaling*, which may be proactive or reactive [109, 113, 116]. *Reactive* autoscaling works based on feedback methods and manages the requirements based on their current state to maintain its performance. *Proactive* autoscaling manages the capacity requirements based on the prediction and assessment of performance in

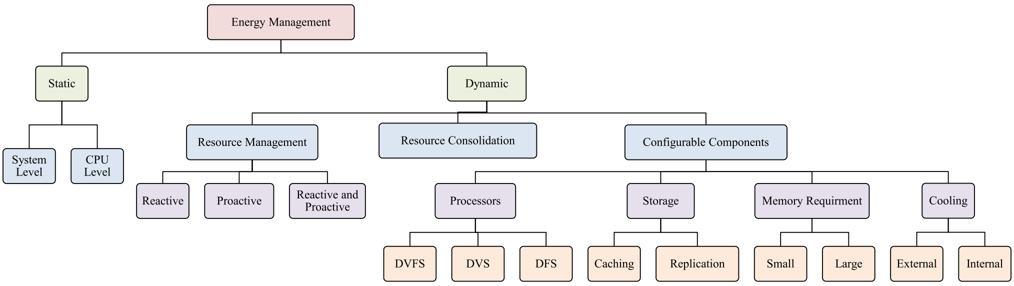

Fig. 4. Taxonomy based on energy management.

terms of QoS values. Based on previous data, predictions have been identified and required action is planned to optimize CDC performance.

3.3.1.5 *Utility Functions.* Latency and cost-based *utility functions* are defined to measure the aspects of capacity planning [113, 114]. *Cost* is defined as the amount of money that can be spent to design a CDC with required configuration. *Latency* is the amount of execution delay with a particular configuration of CDC.

To manage power infrastructure for capacity planning, energy management of resources is required for execution of workloads, which improves the energy efficiency of CDCs.

3.4 Energy Management

Energy management in sustainable computing is an important issue for cloud service providers. Ficco and Rak [5] reported that more than 2.4% of electricity is consumed by CDCs, with a large economic impact of $30 billion globally. The energy requirement to manage the CDCs is also rising in proportion to the operational cost. IBM spends 45% of total expenses on CDC electricity bills and the consumption of electricity will be increased to 101.5 billion kWh by 2022 [10]. Sustainable cloud services are attracting more cloud customers and making it more profitable [62, 60, 64, 65]. Improving energy use reduces electricity bills and operational costs to enable sustainable cloud computing. The essential requirements of sustainable CDCs are optimal software system design, optimized air ventilation, and installing temperature monitoring tools for adequate resource utilization, which improves energy efficiency [67, 74, 120]. There are mainly three levels where energy consumption can be optimized: software level (efficient use of registers, buffers, etc.), hardware level (transistors, voltage supply, logical gates, and clock frequency) and intermediate level (energy-aware resource provisioning techniques) [80]. The evolution of energy management techniques (see Figure 23) and their comparison along with open research challenges [3, 25, 60, 61, 62, 66, 75, 76, 41, 60, 67, 125, 83, 74, 80] are presented in Table 13 of Appendix C.4.

3.4.1 *Energy Management-Based Taxonomy.* Energy management has two important components (static and dynamic), as shown in Figure 4. These taxonomy elements are discussed below, accompanied by relevant examples. The comparison of existing techniques (discussed in Appendix C.4) based on our energy management taxonomy is given in Table 14 of Appendix C.4.

3.4.1.1 *Static Energy Management.* Static energy management is a more engineering-oriented approach, in which circuitry systems are considered more by offline energy management [60]. During design time, the entire optimization happens at system level and deals with factorization, path balancing, transistor sizing, instruction sets, redesigning of architectures, circuit manipulation, and processing centers [67].

Low-power use components are employed in this management approach to reduce energy consumption as much as possible. Static energy management performs at two levels: system level and CPU level. Existing studies [125, 83] found that the *CPU* offers a big scope of optimization of energy consumption because computing components of the CPU consumes 35% to 40% of energy [3, 68, 69, 71]. The optimization of energy at *CPU level* can be performed at the instruction set level or register level. Researchers designed different instruction set architectures to improve resource utilization, such as reduced bit-width architecture at the instruction set level [70]. On the other hand, the activities of the register transfer level are optimal for decreasing energy consumption. Figure 24 of Appendix C.4. shows the energy cost and carbon dioxide emission for static and dynamic energy management techniques [122, 144, 145].

Other components of the *system* besides the CPU that consume large amounts of energy are software systems, network facility, and memory components [143]. Researchers proposed different management techniques to optimize the power consumption of these components based on the setup techniques used in system design [73]. At design time, it is very difficult to select the right components to design a cloud system with maximum synchronization among the components [61, 62, 66]. Other important challenges during system design can be: (i) type of application and software, (ii) selection of operating system, and (iii) placement of servers to reduce delay. Gordan and Fast Array of Wimpy Nodes (FAWN) [3] architecture has been designed to improve the performance of cloud systems by balancing the input-output activities and computation processes by coupling datacenter powering systems and local flash storage with low-power CPUs. Energy consumption can be reduced by proper distribution of resources geographically and the selection of suitable network topologies and components with maximum compatibility [72].

3.4.1.2 *Dynamic Energy Management.*

Software-based policies are used in dynamic energy management to improve energy utilization. There is a different dynamic power range for every component. During low-activity modes, a CPU consumes 30% of the peak value of its energy consumption and can be scaled up and down up to 70% [80]. The dynamic range of energy consumption for disk drives is 50%, 25% for memory, and 15% for network devices such as routers and switches [83]. To improve energy utilization, the number of components can be scaled up or down based on the range of dynamic power. Dynamic energy management is divided into three categories based on the reduction of the dynamic power range: (i) configurable components, (ii) resource consolidation, and (iii) resource management.

Configurable components include the *CPU,* which supports low-activity modes at the component level. Dynamic energy management can be used to control the CPU. The CPU is the main source of energy consumption. Thus, existing research work mainly focused on optimization of energy consumption by the CPU or processor and memory. There is a relationship between power supply, voltage, and operational frequency [66, 62]: ($Power_{Dynamic} = Utilization_{CPU} \times$ Frequency $\times Voltage^2$). Based on the different values of voltage and operational frequency, a CPU can run in different activity modes or C-modes in advance processor architectures. As supply voltage increases, the energy consumption increases quadratically in Complementary Metal Oxide Semiconductor (CMOS) circuits [3]. The values of linear relations can be exploited by changing operation frequency (DFS), voltage (DVS) or both simultaneously (DVFS) [60]. DVFS for energy management is described in Appendix C.4.1 and C-states or C-modes for energy management is described in Appendix C.4.2.

There are a number of methods proposed to control energy consumption by scaling down the high voltage supply, but the best way is to exploit the *stall time.* A high amount of clock speed is wasted while waiting for the data because of the speed gap between processor and main memory. Energy may be saved by reducing the processor frequency through manipulation of supply voltage. For different devices, semiconductor chip vendors optimizing energy consumption use different frequency scaling policies. Eight different kinds of operational frequencies are available in Intel's Woodcrest Xeon Processor [3]. Two CPU throttling technologies developed by AMD are PowerNow and CoolnQuiet [3, 125]. Another, the SpeedStep CPU throttling technology, has been developed by Intel to control energy consumption [62]. The *cooling* can be *internal* (fans) or *external* (as discussed in Section 3.7) for a CDC.

The management of *storage* devices such as disk drives is handled by scalable storage systems to reduce energy consumption because disk drives consume significant amounts of energy. The storage of data can be managed using either *replication* or *caching*. Mechanical operations of storage components consume one-third of the total electricity provided to CDCs, and disks also consume one-tenth during standby mode. The need for storage components is increasing by 60% annually [66, 67]; thus, research on energy consumption control is imperative. Disk drives use only 25% of their storage space, which remains underutilized in large CDCs [3, 71]. Power use can be minimized by reducing underutilization by switching off unnecessary disks. Many mechanisms have been proposed to improve the energy efficiency of disk drives [2]. In large-scale CDCs, the *memory* component may be considered to decrease power use, but it is the least addressed component by researchers. Memory consumes 23% of energy to run a specific workload [83, 125]. The dynamic range for memories is 50%, as discussed above; thus, there is a chance to improve energy consumption in this component [61, 62]. DVFS is also applicable to memory components by reducing frequency and voltage. Storage arrays are the most important components of DRAMs in which power consumption can be reduced. It is challenging to develop energy-aware memory components in cloud computing to reduce power consumption without degradation of performance. Also, it is difficult to manufacture energy-efficient memory devices with lower cost. Existing memory management infrastructures can minimize energy consumption up to 70% [6].

Resource consolidation is a technique for effective use of resources (processor, memory, or network devices) to minimize the number of resources and locations of servers that a cloud company requires to serve user requests [71]. A resource scheduler allocates resources to execute workloads dynamically to avoid over utilization and under-utilization of resources. *Resource management* is an significant challenge because of the following factors: (i) heterogenous resources, (ii) varying costs, (iii) applications with varying requirements (compute, data, network, memory), and (iv) user QoS requirements. Effective resource management includes resource allocation, resource scheduling, and resource monitoring to achieve effective utilization of resources [32]. Many issues need to be addressed to achieve this, including the following [32, 71]:

(a) How to allocate the resources in an energy-efficient manner for the execution of workloads
(b) When to migrate workloads from one machine to another to save energy consumption
(c) Which devices need to be switched off to save energy consumption without degradation of performance

Based on existing research, the techniques above addressed issues to improve energy utilization. These techniques are classified into the following categories: (a) Proactive, (b) Reactive, and (c) Proactive and Reactive. *Proactive* management manages the resources based on the prediction of future performance of the system instead of its current state. The resources are selected based on the previous executions of the system in terms of reliability, energy consumption, throughput, and the like. The predictions are required to be based on previous data and appropriate actions are formulated to optimize energy consumption during resource execution. *Reactive* management works based on feedback methods and manages the resources based on their current state to optimize energy. There is a need of continuous monitoring of resource allocation to find whether

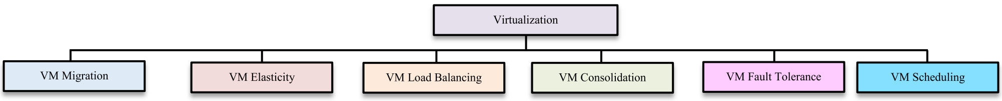

Fig. 5. Taxonomy based on virtualization.

the energy is consumed less than its threshold value or not (threshold value can be based on energy as well as resource utilization). If power usage is higher than threshold value, then corrective action will be taken to optimize the energy consumption. The accuracy of the monitoring module improves the productivity of reactive management. In the case of underutilization of resources, energy consumption can be reduced through VM consolidation or migration as discussed in Section 3.5. Increase in energy consumption also requires effective cooling management because temperature increases due to large amounts of heat. *Reactive and proactive* management manages the resources with minimum value of power usage and maximum value of resource utilization to handle every situation by (i) monitoring the resource execution continuously and (ii) performing the actions based on predicted failures. In real time, it is challenging to accurately forecast the behavior of a system in proactive management. In reactive management, there is a larger overhead, which causes unnecessary delay as well as energy inefficiency [60].

A virtualization technology reduces the number of physical machines or resources and executes the workloads using virtual resources, which leads to a reduction in energy consumption.

3.5 Virtualization

Virtualization technology is an important part of sustainable CDCs to support energy-efficient VM migration, VM elasticity, VM load balancing, VM consolidation, VM fault tolerance, and VM scheduling [88]. Operational costs can be reduced by using VM scheduling to manage cloud resources using efficient dynamic provisioning of resources [102]. During the execution of workloads, VM load balancing is required to balance the load effectively owing to decentralized CDCs and renewable energy resources. Owing to the lack of on-site renewable energy, VM techniques migrate the workloads to the other machines distributed geographically. VM technologies also offer migration of workloads from renewable energy-based CDCs to the CDCs using the waste heat at another site [105]. To balance the workload demand and renewable energy, VM-based workload migration and VM consolidation techniques provide virtual resources using few physical servers. VM fault tolerance creates and maintains the identical secondary VM for the replacement of the primary VM in a failover situation without affecting the availability of cloud service. VM elasticity maintains the performance of the computing system by providing the dynamic adaptation of computing resources or capacity to fulfill the changing requirements of workloads. Waste heat use and renewable energy resource alternatives are harnessed by VM migration techniques to enable sustainable cloud computing [82, 104]. It is a great challenge for VM migration techniques to improve energy savings and network delays while migrating workloads between resources distributed geographically. The evolution of virtualization technologies (see Figure 25) and their comparison along with open research challenges [40, 63, 159, 160, 162, 157, 158, 161, 163, 88, 101, 99, 102, 105, 106, 156, 37] can be found in Table 15 of Appendix C.5.

3.5.1 Virtualization-Based Taxonomy. Based on the literature, virtualization consists of the following components: VM migration, VM elasticity, VM load balancing, VM consolidation, VM fault tolerance, and VM scheduling, as shown in Figure 5. Each of these taxonomy components are discussed below along with their subcomponents and relevant examples.

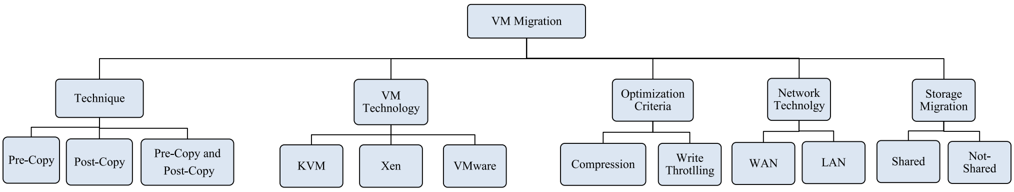

Fig. 6. Taxonomy based on VM migration.

The comparison of existing techniques (discussed in Appendix C.5) based on our virtualization taxonomy is given in Table 16 of Appendix C.5 (VM migration, VM elasticity and VM load balancing) and Table 17 of Appendix C.5 (VM consolidation, VM fault tolerance and VM scheduling).

3.5.1.1 VM Migration-Based Taxonomy. VM migration is a process of relocation of a running VM from one physical machine to another without affecting the execution of user application. Based on the literature [101, 102, 106, 164], VM migration consists of the following components: (i) technique, (ii) VM technology, (iii) optimization criteria, (iv) network technology, and (v) storage migration, as shown in Figure 6.

3.5.1.1.1 Technique. VMs can be migrated from one place to another for better utilization of resources, reducing the under utilization and over utilization of resources [99]. Three types of techniques have been proposed for VM migration: (1) Pre-copy, (2) Post-copy, and (3) Pre-copy and Post-copy. There are two different phases of *pre-copy* technique: warm-up and stop and copy. In the *warm-up* phase, the hypervisor copies the state from the source server to the destination server, which contains the information about the memory state and the CPU state. The *stop and copy* phase copies the pending files (if any file is modified during the warm-up phase) from the source to destination servers and starts the execution at the destination server [88]. In *post-copy*, it stops the VM at the source server, transfers all the details, such as CPU state and memory state, to the destination server, and starts execution. Some VM migration mechanisms use both *pre-copy* and *post-copy* together to transfer states from one server to another.

3.5.1.1.2 VM technology. There are three different types of technology that are available in the literature for VM migration: KVM, Xen, and VMware. *KVM* is a kernel-based VM, which permits many operating systems (OSs) to share a single resource or hardware. *Xen* works based on a microkernel design to share the same resources to run multiple OSs. *VMware* can be used for application consolidation to provide services through virtualization [101].

3.5.1.1.3 Optimization criteria. It has been determined that optimization criteria for virtualization technology can be compressed or go through write throttling. ESXi is an independent hypervisor, which offers memory *compression* cache to increase the performance of VMs and further increases the capacity of the CDC [105]. *Write throttling* is used to perform write and incoming copy operations, which limit the transfer of data [106].

3.5.1.1.4 Network technology. There are two different types of network technologies used for VM migration: WAN and LAN. *Wide Area Network (WAN)* is used to migrate a VM geographically using a wireless connection, while a *Local Area Network (LAN)* is used to migrate a VM from one server to another within a limited area.

3.5.1.1.5 Storage migration. In this technique, storage from one running server to another can be migrated without affecting the workload execution of VMs. Storage migration can also be used

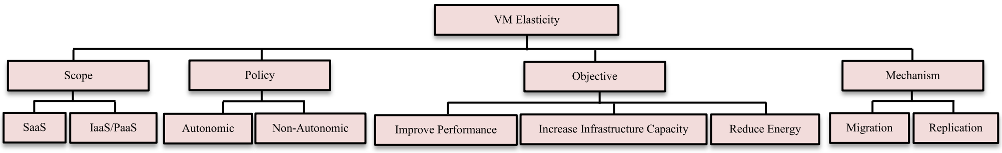

Fig. 7. Taxonomy based on VM elasticity.

to upgrade storage resources or transfer service [101, 32]. The distributed file systems can be used to provide shared storage space.

The main issues with VM migration in geographically distributed datacenters are discussed in Appendix C.5.1. Container as a Service (CaaS) for virtualization is discussed in Appendix C.5.2.

3.5.1.2 VM Elasticity-Based Taxonomy. VM elasticity enables the automatic provisioning and de-provisioning of computing resources to fulfill the changing demand of workloads at runtime. Based on the literature [40, 37, 163, 164], VM elasticity consists of the following components: (i) scope, (ii) policy, (iii) objective, and (iv) mechanism, as shown in Figure 7.

3.5.1.2.1 Scope. It defines the location, where the elasticity actions are managed, which can be application (SaaS) or platform level (PaaS) and infrastructure level (IaaS) [62]. At *IaaS level*, the elasticity controller monitors the application execution and performs different decisions based on resource (hardware) scalability. At *SaaS or PaaS level*, the elasticity controller is implanted in the application or within the execution platform, which performs the dynamic scalability of cloud resources.

3.5.1.2.2 Policy. There are two types of policies for the execution of elasticity actions: autonomic and non-autonomic [66]. In *autonomic* policy, the cloud system or application controls the elasticity actions and performs actions based on the SLA constraints. In *manual* policy, the user monitors the virtual environment and performs the elasticity actions accordingly.

3.5.1.2.3 Objective. VM elasticity techniques have three main objectives: (1) improve performance, (2) increase infrastructure capacity, and (3) reduce energy [67]. The main objective of VM elasticity techniques is to improve *performance*, such as optimal searching of VM and reducing the task rejection rate and makespan. The second objective is to *reduce energy consumption* of CDCs during execution of workloads. The third objective is to *improve the infrastructure capacity* by adding different resources at runtime to execute workloads within their specified budget and deadline.

3.5.1.2.4 Mechanism. There are two different mechanisms for VM elasticity as identified from the literature [83]: migration and replication. *Migration* refers to moving the VM from one physical machine to another for effective use of application load using deconsolidation and consolidation of resources. *Replication* refers to elimination and removal of instances (application modules, containers, VMs) from the virtual environment.

3.5.1.3 VM Load Balancing-Based Taxonomy. VM load balancing refers to the optimization of use of VMs to reduce resource wastage due to underloading and overloading of resources. It helps to achieve QoS and maximize resource use to improve performance of cloud service. Based on the literature [88, 101, 162], VM load balancing consists of the following components: (i) resource-aware, and (ii) performance-aware, as shown in Figure 8.

3.5.1.3.1 Resource-aware. CDCs require different types of resources (memory, processor, cooling, storage, networking etc.) to execute user workloads [74]. *Resource-aware* load balancing

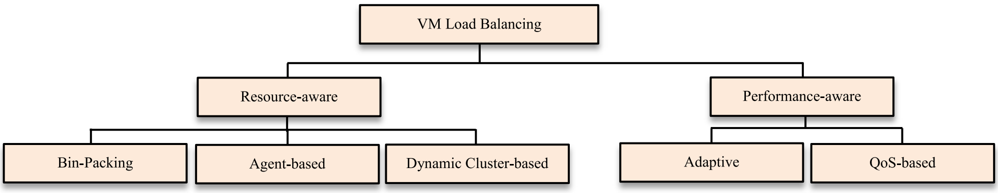

Fig. 8. Taxonomy based on VM load balancing.

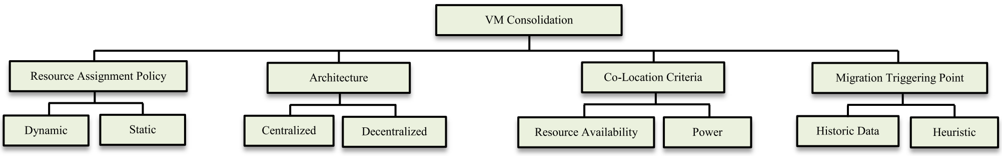

Fig. 9. Taxonomy based on VM consolidation.

algorithms execute workloads, also monitoring and analyzing the different performance parameters related to resources such as energy consumption, degree of resource capacity imbalance, and resource use to perform load balancing. There are three different types of resource-aware load-balancing algorithms: bin-packing, agent-based, and dynamic cluster-based. In *bin-packing*, different bins are used to pack objects of different capacities. Bin packing uses a minimum number of bins to provide the same capacity in a balanced way. In *agent-based*, a software agent is used to monitor the performance of different components, such as network devices, storage devices, and processors, and balances the load effectively. In *dynamic cluster-based*, resources are categorized automatically based on requirements and availability of resources. Further, categorized resources are allocated for execution of workloads with maximum resource utilization and minimum energy consumption.

3.5.1.3.2 Performance-aware. In *performance-aware* load-balancing algorithms, different performance parameters are analyzed to make decisions for effective load balancing of VMs [80]. There are two different types of performance-aware load-balancing algorithms: adaptive and QoS-based. In *adaptive*, performance is maintained using a dynamic computing environment for execution of workloads with changing behavior. In *QoS-based*, resources are provisioned and scheduled for workload execution by fulfilling the QoS requirements of applications such as energy efficiency, makespan, execution cost, and response time.

3.5.1.4 VM Consolidation-Based Taxonomy. VM consolidation refers to the effective use of VMs to improve resource utilization and reduce energy consumption [49]. Based on the literature [99, 102, 159, 160], VM consolidation consists of the following components: (i) resource assignment policy, (ii) architecture, (iii) co-location criteria, and (iv) migration triggering points, as shown in Figure 9.

3.5.1.4.1 Resource assignment policy. This policy defines the mechanism to select resources for VMs within a CDC [125] and can be static or dynamic. In the *dynamic* approach, VMs are reconfigured using dynamic attributes proactively based on the demand of workloads. In the *static* approach, maximum resources are preassigned to a VM for workload execution.

3.5.1.4.2 Architecture. There are two different types of architectures used in VM consolidation techniques: centralized and decentralized, as described in Section 3.1.1.4. There is no risk of a

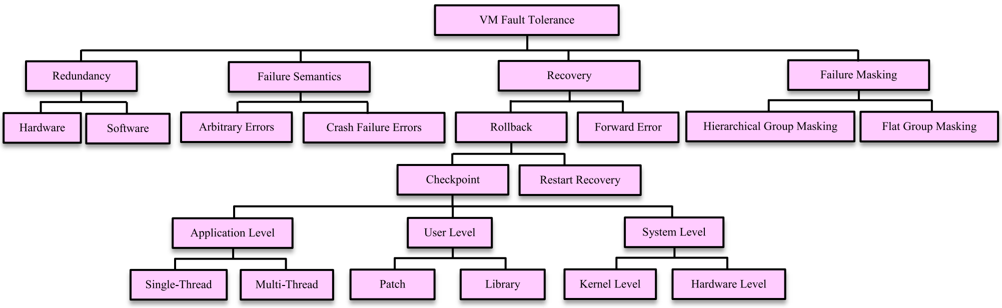

Fig. 10. Taxonomy based on VM fault tolerance.

single failure point in *decentralized* architectures, while *centralized* architectures are prone to a single failure point.

3.5.1.4.3 Co-location criteria. There are two main types of co-location criteria in VM consolidation techniques, which is considered based on resource availability and power [71]. VMs can be co-located from one CDC to another (i) if less resources are *available* in the current CDC or (ii) if there is unavailability of adequate *power* to run the CDC.

3.5.1.4.4 Migration triggering point. VMs can be migrated from one CDC to another for consolidation. The target CDC is identified using two different approaches [67]: historic data and heuristic based. In the *historic data*-based approach, the VM can be migrated to the most efficient CDC based on the historic data of previous performances. In the *heuristic*-based approach, the most efficient CDC can be identified based on performance parameters such as resource utilization, energy consumption, and response time.

3.5.1.5 VM Fault Tolerance-Based Taxonomy. VM fault tolerance supports the primary VM by maintaining the identical secondary VM to provide continuous availability of cloud service in case of VM failure. Based on the literature [105, 106, 157, 161], VM fault tolerance consists of the following components: (i) redundancy, (ii) failure semantics, (iii) recovery, and (iv) failure masking, as shown in Figure 10.

3.5.1.5.1 Redundancy. In the case of resource failure, redundancy provides redundant components to maintain the performance of the computing system, which can be software or hardware [125]. For *hardware* components, the physical redundancy technique adds redundant hardware components to tolerate failures, which support the computing system to continue its service in an efficient manner. For *software* components, two different types of processes are created: active (primary) and passive (backup). The backup process is identical to the primary process; the backup process will be active during the failure of the primary process to maintain the performance of the system.

3.5.1.5.2 Failure semantics. This refers to the selection of failure tolerance method based on the two types of failure modes [83, 100]: arbitrary errors and crash failure errors. An *arbitrary error* occurs when a communication service loses or delay messages or messages may be corrupted. A *crash failure error* occurs when a system suddenly stops processing of instructions. To deal with both type of failures, a computing system needs a duplicate processor.

3.5.1.5.3 Recovery. This mechanism replaces the erroneous state with a stable state using different recovery mechanisms [103, 122]: Forward Error Recovery (FER) and Backward Error Recovery

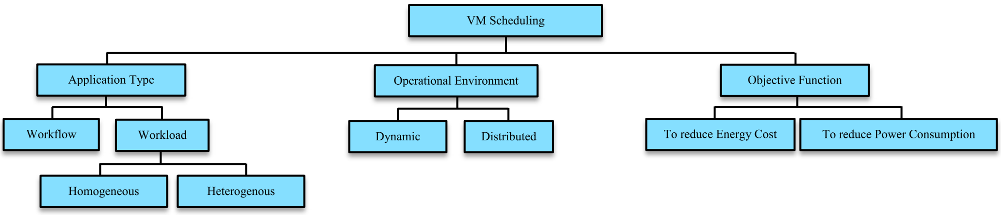

Fig. 11. Taxonomy based on VM scheduling.

(BER, or rollback). The *FER mechanism* tries to correct the errors to move the system into a new correct state and this mechanism is effective when there is a need of continue service. BER or rollback recovery is a widely used fault tolerance mechanism, which consists of two different methods: checkpoint and restart recovery. The *restart recovery* mechanism performs the process of rebooting to recover or restore the system to its correct state. To incorporate fault tolerance into the system, a snapshot of the application's state is saved so that the system can reboot from that point in case of a system crash; this process is called checkpointing. *Checkpoints* can be performed at three different levels: application, user, and system. At the *application level*, a checkpointing code is inserted automatically into the application code if failure has occurred. The checkpointing code can be written using single-thread or multi-thread programming. At the *user level*, an application program is linked to the *library*; Condo [20] and Esky [66] are library implementations. Further, the user can use *patch* to perform user-level checkpointing. At the *system* level, the process of checkpointing can be performed at the OS kernel level and hardware level. The digital hardware is used in *hardware level checkpointing* to modify a group of commodity hardware. *OS kernel* level checkpointing installs the available package for a particular OS.

3.5.1.5.4 Failure masking. The failure masking technique ensures the availability of cloud service during node failure without the user observing any interruption [144, 145]. There are two types of masking techniques: flat and hierarchical group masking. In *flat group masking*, individual workers are appearing as a single worker and hidden from the clients, and a new worker will be selected using a voting process [14] in the case of failure. In *hierarchical group masking*, a central coordinator controls the activities of different workers; the coordinator selects the new worker in the case of failure.

3.5.1.6 VM Scheduling-Based Taxonomy. The VM scheduling algorithm schedules the virtual resources (local or remote) effectively for workload execution. Based on the literature [60, 61, 40, 107, 157, 158], VM scheduling consists of the following components: (i) application type, (ii) operating environment, and (iii) objective function, as shown in Figure 11.

3.5.1.6.1 Application type. Cloud application consists of two different tasks, which need computing resources for their execution [95]: workload and workflow. A *workload* is the execution of a set of instances to achieve desired output and it can be either *homogeneous* (same QoS requirements) or *heterogenous* (different QoS requirements). *Workflow* is a combination of interrelated tasks, which distribute on different resources to achieve a single objective.

3.5.1.6.2 Operational environment. There are two types of operational environments: dynamic and distributed, or an environment can be both [143]. In a *dynamic* environment, VMs are scheduled for workload execution to reduce resource waste and energy consumption. In a *distributed* environment, optimized VMs are scheduled from different CDCs, which are distributed geographically to improve resource utilization for workload execution.

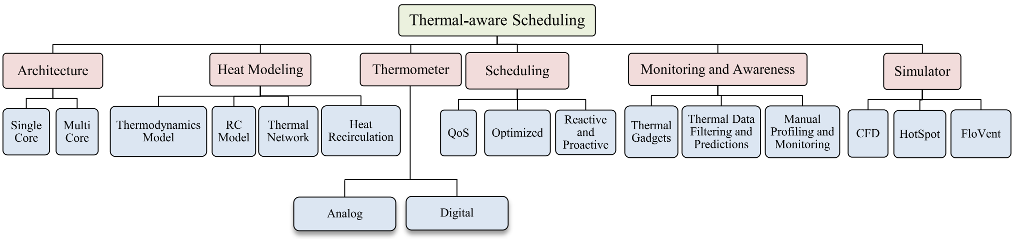

Fig. 12. Taxonomy based on thermal-aware scheduling.

3.5.1.6.3 Objective function. The literature has reported that there are two types of objective functions for VM scheduling: (1) to reduce energy cost and (2) to reduce power consumption. The *energy cost* is a combination of monetary and non-monetary costs associated with energy use for scheduling VMs [3]. The *power consumption* is the amount of electricity expended by a resource to complete the execution of an application [62].

For effective management of virtualized CDCs, thermal-aware scheduling is required to execute workloads on energy-efficient computing resources, which further reduce the heat recirculation and therefore the load on the cooling systems.

3.6 Thermal-Aware Scheduling

CDCs consist of a chassis and racks to place the servers to process the IT workloads. To maintain the temperature of datacenters, cooling mechanisms are used to reduce heat [86]. Thus, there is a need for effective management of temperature to run the CDC efficiently. Servers produce heat during execution of IT workload; thus, cooling management is required to keep room temperature stable [92]. The processor is an important component of a server and consumes the most electricity. Sometimes the heat generation of processors is higher than the threshold because servers are organized in a compact manner [93]. Both cooling and computing mechanisms consume a huge amount of electricity. It would be better to reduce the energy consumption instead of improving the cooling mechanism [90]. To solve the heating problem of CDCs, thermal-aware scheduling is designed to minimize cooling setpoint temperature, hotspots, and thermal gradients. Thermal-aware scheduling is better than heat modeling [142]. Thermal-aware scheduling based on heat modeling performs computational scheduling of workload. Thermal-aware monitoring and profiling module monitors and assess the distribution of heat in CDCs while profiling maintains the details of computational workload, microprocessors, and heat emission of servers. With the use of renewable energy, the load of cooling can be decreased to enable sustainable CDCs. The evolution of thermal-aware scheduling techniques (see Figure 26) and their comparison along with open research challenges [50, 53, 92, 93, 86, 94, 97, 98, 91, 85, 89, 90] can be found in Table 18 of Appendix C.6.

3.6.1 Thermal-Aware Scheduling-Based Taxonomy. The components of thermal-aware scheduling are (i) architecture, (ii) heat modeling, (iii) a thermometer, (iv) scheduling, (v) monitoring and awareness, and (vi) a simulator, as shown in Figure 12. Each of these taxonomy elements are discussed below, along with relevant examples. The comparison of existing techniques (discussed in Appendix C.6) based on our thermal-aware scheduling taxonomy is given in Table 19 of Appendix C.6.

3.6.1.1 Architecture. Thermal-aware scheduling techniques have been designed based on two different architectures: single core and multi-core [86, 92, 93]. Thermal-aware scheduling techniques execute workloads based on their priorities at different processor speeds for *single-core* architecture, and for execution of a high-priority workload, the current workload can be preempted. Generally, high-priority workloads are running at high speed and the temperature of the processor can reach its threshold value. To optimize the temperature of the processor, low-priority workloads are running at a lower speed to cool down the processor. To improve the execution of thermal-aware scheduling, a *multi-core* processor is used, in which a task is divided into a number of threads and independent threads are running on different cores based on their priorities. Multi-core processors are designed with thermal-aware aspects such as intelligent fan control, clock gating, and frequency scaling. These aspects are working in coordination to control the temperature within its operating limits. If one core is getting hot, then a thread can be transferred to another cooler core to maintain the temperature.

3.6.1.2 Heat-modeling. It is an effective mechanism in thermal-aware scheduling to develop a relationship between eventual heat dissipation and energy consumed by computing devices. The scope of heat models is defined based on evaluation of environmental variables such as temperature, air pressure, and power. The selection of heat model also affects energy efficiency. The types of heat models used in the literature are (i) the thermodynamics model, (ii) RC model, (iii) thermal network, and (iv) heat recirculation. The *thermodynamics model* is used to explore the heat exchange mechanisms in CDCs. The value of heat is quantified using the law of energy conservation [89, 90]. The thermodynamic process produces the details of heat emissions and energy consumption and passes cold air to remove heat from the datacenter. Researchers are still working on this process for further optimization. The *RC model* is basically a resistor–capacitor (RC) circuit that forges a relationship between electrical phenomena of the RC circuit and heat transfer. Temperature difference between two surfaces and energy consumption is used to determine the value of R and C for conductance and convention. The value of RC is not changed after manufacturing the processor package. The RC model is used to determine the value of various thermal parameters. The *thermal network* is based on both the RC model and thermodynamics model. In a thermal network, every node of a CDC belongs to one of the networks, which can be an IT network or cooling network. A server executes a workload by consuming energy and producing heat and the server is part of both the cooling and IT networks. The thermal network is efficient for heat modeling of heterogenous equipment of a datacenter. *Heat recirculation* deals with mixing hot air (coming from server outlets) and cold air (coming from the cooling manager). The temperature of cold air is changing with time after entering into CDCs. To maintain the temperature of a CDC, it is a great challenge to provide the uniform cold air temperature every time. The resource utilization of servers that participates in heat recirculation will be reduced and performance of CDCs is also affected in terms of QoS.

3.6.1.3 Thermometer. This is a device that is used to measure the temperature of CDCs. Two types of thermometers have been identified from the literature [14, 86, 92]: digital and analog. The *digital* or infrared thermometer is an electronic device that uses a digital sensor to provide a digital display. Most digital thermometers are resistive thermal devices that uses a function of electrical resistance to measure temperature variations. The *analog* thermometer contains alcohol, which falls or rises as it contracts or expands with temperature variations. Temperature value is displaying in degrees Celsius or Fahrenheit, which is marked on a glass capillary tube.

3.6.1.4 Scheduling. The energy consumed by CDCs is used for execution of workloads, but it is dissipated as heat. Lower energy is used to remove heat while workloads are scheduling using thermal-aware aspects. Thermal profiles of thermal-aware schedulers are used to determine the resource with minimum dissipation of heat in CDCs. The aim of thermal-aware scheduling is to reduce dissipation of heat from active servers and minimize the active servers by turning off idle servers. Three types of thermal-aware scheduling used in the literature [85, 86, 89, 92] are (i) QoS, (ii) optimized, and (iii) reactive and proactive. *QoS-based thermal-aware scheduling* schedules the energy-efficient resources to improve the performance of the CDC. The scheduler controls the temperature and reduces the load of overcooling using dynamic thermal management techniques. Further, a challenge of maintaining the SLA based on these QoS parameters is introduced and requires the trade-off between cost saving and compensation or penalty in the case of SLA violations. *Optimized thermal-aware scheduling* schedules workloads using the concept of autonomic computing. These techniques are basically a combination of heat-recirculation and thermal-aware techniques. The main aim of server-based scheduling techniques is to reduce the peak inlet temperature, which is increased by heat recirculation. Heat recirculation can be minimized by placing lesser workloads on servers that are nearer to the floor. Processor-based scheduling techniques execute the workloads by sustaining the steady core temperature, called *throttling*. Earlier, workloads are executed using zig-zag schemes till a temperature threshold is achieved. *Reactive* management works based on feedback methods and manages the temperature based on their current state to maintain its temperature. Continuous monitoring of thermal-aware scheduling is needed to determine whether the temperature is lower than its threshold value or not. If the temperature is higher than a threshold value, then corrective actions will be taken to make it stable. The *proactive* approach manages the resources based on the prediction and assessment of temperature and thermal profiling. Based on previous data, predictions have been identified and required action is planned to reduce temperature during scheduling.

3.6.1.5 Monitoring and Awareness. Thermal monitoring and awareness is used to perform thermal-aware scheduling decisions. The thermal profile is created based on resultant heat dissipation and power consumption for thermal awareness, which is used to rank the servers for future scheduling decisions. There are three different methods of thermal monitoring and awareness, as identified from the literature [14, 32, 85, 86]: (i) manual profiling and monitoring, (ii) thermal gadgets, and (iii) thermal data filtering and predictions. In *manual profiling and monitoring*, heat generation and recirculation and power consumption of individual servers are noted manually to create a thermal profile. If there are no real data available, then simulation tools can be used for manual profiling. Some thermal-aware scheduling techniques [89, 92, 93, 97] estimate the thermal index to evaluate the efficiency of different CDCs and perform their ranking. *Thermal gadgets* such as thermal cameras and sensors are used to generate accurate and timely thermal information automatically. Multiple sensors can be used per unit area and both onboard and external thermal sensors can be used to collect thermal information. In *thermal data filtering and predictions*, a rise in temperature and resulting heat can be predicted for proactive thermal-aware scheduling, which helps to make effective decisions to minimize thermal gradient and peak outlet temperature. The advance prediction of temperature and heat can help to maintain the QoS during workload execution.

3.6.1.6 Simulator. The results of thermal simulators can be used to create thermal profiles. There are three different simulators identified from the literature [14, 86, 92, 14, 32, 85, 86]: (i) CFD, (ii) HotSpot, and (iii) FloVent. The *Computational Fluid Dynamics (CFD)* simulator is used to analyze and optimize airflow and heat transfer for CDCs to create the thermal profile, which further helps to create a thermal map. *HotSpot* is a temperature modeling tool [14], which uses thermal resistances to design the architecture of CDCs based on power density and hence cooling costs, which are rising exponentially. The *FloVent* simulator [14] is used to predict contamination distribution, heat transfer, and 3D airflow for different types of CDCs, which mainly focuses on air conditioning and ventilating systems.

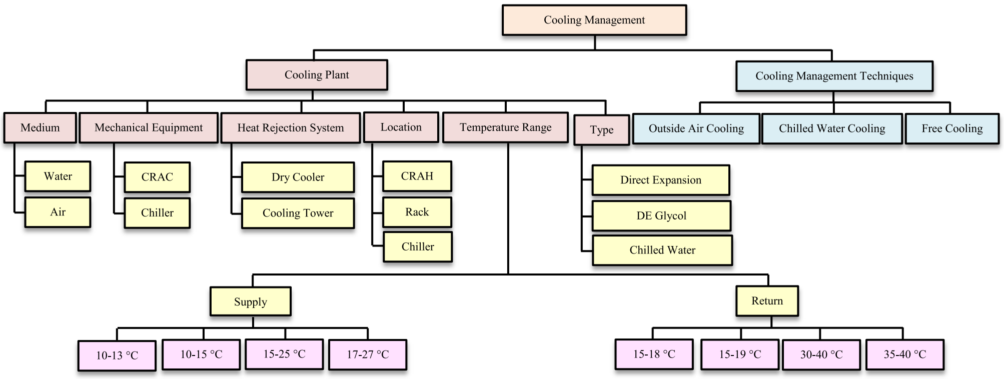

Fig. 13. Taxonomy based on cooling management.

Effective cooling mechanisms are needed to maintain the temperature of CDCs to enable sustainable cloud computing.

3.7 Cooling Management

The increasing demand for computation, networking, and storage expands the complexity, size, and energy density of CDCs exponentially, which consumes a large amount of energy and produces a huge amount of heat [14]. To make CDCs more energy efficient and sustainable, we need an effective cooling management system, which can maintain the temperature of CDCs [21]. Heat dissipation is a critical factor to be considered for cooling management of CDCs, which affects the reliability and availability of the cloud service. In cloud datacenter CDCs, high heat density causes high temperature, which needs to be controlled for smooth functioning of CDCs [86]. Effective cooling management can attain complete environmental control, including pollution concentration, humidity, and air temperature [92]. Thus, it is necessary to discuss the existing and emerging technologies for datacenter cooling systems to determine the effective approach to maintaining CDCs working in a safe and reliable manner. The evolution of cooling management techniques (see Figure 27) and their comparison along with open research challenges [150-155] are provided in Table 20 of Appendix C.7.

3.7.1 Cooling Management-Based Taxonomy. Based on the literature [150-155], cooling management consists of the following components: (i) cooling management techniques and (ii) the cooling plant as shown in Figure 13. Each of these taxonomy elements are discussed below along with relevant examples. The comparison of existing techniques (discussed in Appendix C.7) based on our cooling management taxonomy is given in Table 21 of Appendix C.7.

3.7.1.1 Cooling Plant. The cooling plant is a system that provides cooling to space where the CDC is placed and consists of the following components: (i) medium, (ii) mechanical equipment, (iii) a heat rejection system, (iv) location, (v) type, and (vi) temperature. The cooling system uses two different types of mediums to produce cooling: water and air. The *water-based* cooling system uses a water pumping mechanism to generate cooling, while the *air-based* cooling system uses an air compressor mechanism to produce cooling. *Mechanical equipment* is used to maintain the humidity, air distribution, and temperature in the CDC. Two different types of mechanical equipment are used in cooling systems: Computer Room Air Conditioning (CRAC) and chiller. The *Heat Rejection System (HRS)* performs the process of heat removal via two methods: dry cooler and cooling tower. Different types of temperature range are established for different locations in cooling systems [14, 86, 92, 150, 155]; location can be (1) chiller, (2) rack, and (3) Computer Room Air Handler (CRAH). There are two types of temperature classification for three different locations with different temperature ranges: (1) supply temperature and (2) return temperature. The different *types* of cooling plants are (1) Direct Expansion (DE) air-cooled systems, (2) DE glycol-cooled systems, and (3) chilled water systems [152, 154]. The DE air-cooled system contains CRAC and an air-cooled condenser as a HRS. In DE glycol-cooled systems, a glycol mixture is used as heat transfer fluid from the CRAC to the dry cooler. In the chilled water system, a chiller provides cold water to the CRAH.

3.7.1.2 Cooling Management Techniques. The literature [14, 86, 92, 150, 155] identified three different types of cooling management techniques: (i) outside air cooling, (ii) chilled water cooling, and (iii) free cooling. In *outside air cooling*, the cooler is used to bring the fresh air from outside and cooled and pushed it through the CRAC, which is better than an air recirculation mechanism. In the *chilled water cooling* system, electricity is used to freeze water at night and circulate this water throughout the CRAC unit during the day. In *free cooling*, air is passed into a chamber, which performs cooling through water evaporation [14].

There is a need to maximize the use of renewable energy for cooling, which further reduces carbon footprints and environmental problems.

3.8 Renewable Energy

Sustainable computing needs energy-efficient workload execution by using renewable energy resources to reduce carbon emissions [117]. Fossil fuels such as oil, gas, and coal generate brown energy, which produces carbon-dioxide emissions in large quantities. Green energy resources such as sun, wind, and water generate energy with nearly zero carbon-dioxide emissions [121]. One type of green energy is hydroelectricity, which is produced using hydraulic power. Wind and solar energy can be purchased from off-site companies or can be generated using on-site equipment [118]. In the next decade, the cost/watt will be reduced by half for renewable energy due to following: (i) government organizations provide monetary incentives for the incorporation of resources of renewable energy, (ii) the storage capacity of rechargeable batteries will be increased, and (iii) advancement in technology to improve capacity of materials such as photovoltaic arrays [124]. Workload migration and energy-aware load-balancing techniques addressed the issue of unpredictability in the supply of renewable energy. To achieve 100% availability of cloud services, adopting hybrid designs of energy generation is recommended, which use energy from renewable resources and grid resources [117]. Mostly, sites of commercial CDCs are located away from abundant renewable energy resources. Consequently, portable CDCs are placed nearer to renewable energy sources to make them cost-effective. Dynamic load-balancing technique and renewable energy-based workload migration are discussed in Appendix C.8. The evolution of techniques for renewable energy (see Figure 28) and their comparison along with open research challenges [117-119, 121, 124, 126, 148] can be found in Table 22 of Appendix C.8.

3.8.1 Renewable Energy-Based Taxonomy. Based on the literature [8], renewable energy consists of the following components: (i) workload scheduling, (ii) focus, (iii) source of energy, (iv) location-aware and (v) storage devices, as shown in Figure 14. These taxonomy elements are discussed below along with relevant examples. The comparison of existing techniques (discussed in Appendix C.8) based on our renewable energy taxonomy is given in Table 23 of Appendix C.8.

3.8.1.1 Workload Scheduling. The scheduling of workloads in renewable energy-aware techniques has been done in two ways: (i) dynamic load balancing and (ii) power preserving. *Dynamic load balancing* is the most well-known approach to make a balance between renewable energy

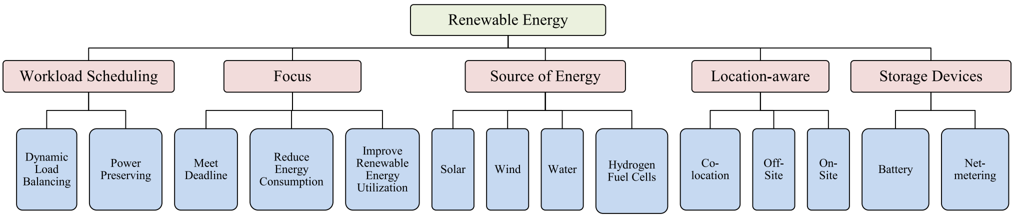

Fig. 14. Taxonomy based on renewable energy.

and grid energy. These techniques supply renewable energy to execute workloads efficiently and predict the amount of energy that can be produced to run a CDC and the amount of energy that is needed to execute the workloads using that energy at the demand side.

There is a great need for renewable energy for deadline-oriented workloads [121, 124]. On the other side, workloads are scheduled using server *power-preserving* techniques. These techniques use power transition and voltage scaling to run a suitable workload using available renewable energy to balance demand-supply. Further, DVFS-based power-preserving techniques are also designed to control the energy based on operational frequency along with voltage scaling. In these approaches [117, 118, 119, 123], the workload (web application) requests are distributed to the specific datacenter by matching the workload demand with the available renewable energy across geo-dispersed CDCs by using a load-balancing algorithm. This approach follows two levels of load balancing: (i) at the local level, redirecting the request within web servers in a datacenter, known as local load balancing; and (ii) at the global level, redirecting the requests among local load balancers related with a CDC, known as global load balancing. Each datacenter has an autoscaler in addition to a local load balancer that adds/removes web servers dynamically in response to the request [124]. The incoming request is distributed among a geo-dispersed CDC based on the place that has a higher availability of renewable energy so that maximum renewable energy is used for making the datacenter sustainable. In the case of not having enough renewable energy, the request is redirected to the location having cheap brown electricity. The global load balancer, uses a "weighted round robin" load-balancing algorithm to redirect the requests.

3.8.1.2 Focus. There are three main objectives of renewable energy-aware techniques, to (i) meet deadline, (ii) reduce energy consumption, and (iii) improve renewable energy utilization [121, 126, 139]. The SLA is an important component and workload should be executed without violation of the SLA. Cloud providers are mainly focused on the *deadline* of the workload during execution. Other renewable energy-aware techniques focus on minimizing *power usage* of CDCs to execute workloads. Further, renewable energy can be used effectively while placing the CDC nearer to the source of *renewable energy* to save more energy and used to process more work.

3.8.1.3 Source of energy. There are four different kinds of energy sources as identified from the literature [118, 117]: (i) solar, (ii) wind, (iii) water, and (iv) hydrogen fuel cells. The renewable energy can be generated using sunlight or it can be generated using *wind* to run a generator to produce electricity. Some techniques use the combination of both solar and wind. Other sources of renewable energy can be *water* as well as *hydrogen fuel cells* [117, 126].

3.8.1.4 Location-aware. In renewable energy generation, energy can be stored using three different localities [121, 124, 127]: (i) on-site, (ii) off-site, and (iii) co-location. In an *on-site* locality, use of renewable energy is done at the same place where energy is produced. *Off-site*, the place of renewable energy use is different than the place generating energy, which means that energy can

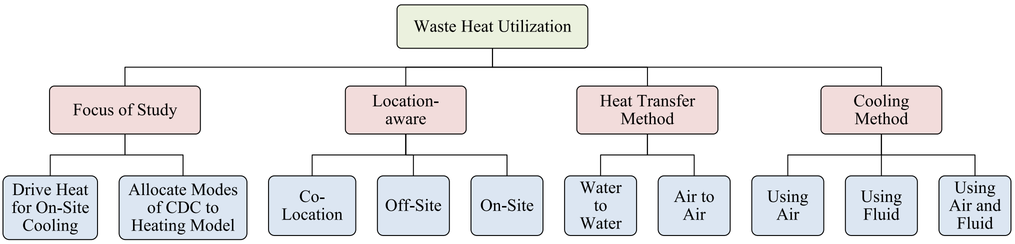

Fig. 15. Taxonomy based on waste heat utilization.

be transported to an off-shore site. On the other hand, CDCs are *co-located* from different places to sites where the chances of renewable energy use exist.

3.8.1.5 Storage devices. There are two main storage devices used by renewable energy-aware techniques to store energy [118, 119, 128]: battery and net-metering. Lithium ion *batteries* are using to store energy effectively. *Net-metering* is another device that can be used to store generated energy for the future. Waste heat can be another source of renewable energy, which can be used in an efficient manner that generates electricity or can be used for heating houses and greatly reduce electricity costs and carbon emissions.

3.9 Waste Heat Utilization

Reuse of waste heat is becoming a solution for fulfilling energy demand in energy conservation systems because fossil fuel deposits are quickly dwindling. Cooling management is necessary to maintain the temperature of CDCs in operational range due to generation of large amounts of heat during energy consumption. The cooling mechanism of CDCs consumes large amounts of electricity: 40% to 50% [3, 71]. Power densities of servers are increased by using stacked and multicore server designs, which further increases cooling costs. The energy efficiency of CDCs may be improved by reducing the energy used in cooling. There is a need to change the location of CDCs to reduce cooling costs, which can be done through placing the CDCs in an area that has free cooling resources. Due to consumption of large amounts of energy, CDCs are acting as a heat generator [129, 130]. The vapor-absorption-based cooling systems of CDCs can use waste heat, then remove the heat while evaporating. Vapor-absorption-based free cooling mechanisms can make the value of PUE ideal by neutralizing cooling expenses. Low-temperature areas can use the heat generated by CDCs for heating facilities. The literature reports [3, 131] that there are two main solutions to control the temperature of CDCs: (1) relocation of CDCs to nearby waste heat utilization recovery places, and (2) vapor-absorption-based cooling systems. The two waste heat utilization techniques—Air Recirculation and Power Plant Co-location—are discussed in Appendix C.9. The evolution of waste heat utilization techniques (see Figure 29) and their comparison along with open research challenges [130-134, 147] can be found in Table 24 of Appendix C.9.

3.9.1 Waste Heat Utilization-Based Taxonomy. Based on the literature, waste heat utilization consists of the following components: (i) focus of study, (ii) location-aware, (iii) heat transfer method, and (iv) cooling method, as shown in Figure 15. These taxonomy elements are discussed below along with relevant examples. The comparison of existing techniques (discussed in Appendix C.9) based on our waste heat utilization taxonomy is given in Table 25 of Appendix C.9.

3.9.1.1 Focus of study. Existing waste heat utilization techniques focus on two different ways to utilize heat: (i) vapor-absorption-based cooling systems and (ii) give heat to co-located datacenter buildings [130, 131]. The first way is utilizing heat for *on-site cooling* using vapor absorption, in which heat is generated during the execution of workloads. The second way is to distribute the heat generated from CDCs to the *heating model* using different modes of transfer. Heat modeling is an effective mechanism in thermal-aware scheduling to develop a relationship between eventual heat dissipation and energy consumed by computing devices.

3.9.1.2 Location-aware. In waste heat utilization, heat can be recovered using three different localities: (i) on-site, (ii) off-site, and (iii) co-location, as described in Section 3.8.1.4.

3.9.1.3 Heat transfer method. There are two different methods available for transferring heat: (i) water to water and (ii) air to air [133]. The *water-to-water* heat transfer method is based on a refrigerator mechanism, in which heat is transferred from the source side to the load side using conditioned fluid (hot or cold). Boiler or cooler can be used at both sides of the exchanger based on the purpose of the transfer. The *air-to-air* heat transfer method is based on vapor compression refrigeration, which uses reverse-cycle air conditioners to transfer heat from one place to another.

3.9.1.4 Cooling method. As identified from the literature, there are three types of cooling methods used in existing waste heat utilization techniques: (i) using air, (ii) using water, and (iii) using air and water [132, 134]. An evaporative cooler is a device that uses evaporation of water to cool *air* and it is based on vapor-compression refrigeration cycles. On the other hand, the cooling effect is produced by consumption of *water* through evaporation. Both water- and air-based cooling mechanisms are used by WHU techniques.

4 OUTCOMES

The outcomes of this systematic review are discussed in Appendix D.

5 OPEN CHALLENGES AND FUTURE DIRECTIONS: A SUMMARY

We surveyed 142 research papers in this systematic review and presented them in a categorized manner. The focus of our systematic review is broader than the existing surveys, as discussed in Table 1 of Appendix A. This survey used methodical survey technique to conduct a systematic review and comprises the most recent research related to sustainable cloud computing. In addition to the nine categories of sustainable cloud computing, we covered the other research issues related to the sustainability of emerging technologies, such as Internet of Things and smart cities. A systematic methodology has been used to develop an evolution of categories of sustainable cloud computing that identifies optimization parameters, metrics, open issues, and Focus of Study (FoS). We explored and compared the existing techniques based on the proposed taxonomy. We documented the research issues addressed and open challenges that are still unresolved in sustainable cloud computing and discussed in Appendix E.

5.1 Open Challenges

The identified various open challenges of sustainable cloud computing are discussed in Appendix E.1.

5.2 Implications for Research and Practice

The implications for research and practice are discussed in Appendix E.2.

5.3 Integrated: Sustainability vs. Reliability

The trade-off between sustainability and reliability is discussed in Appendix E.3.

5.4 Emerging Trends and Their Impact

The emerging trends and their impact are discussed in Appendix E.4.

6 SUSTAINABLE CLOUD COMPUTING ARCHITECTURE: A CONCEPTUAL MODEL

The conceptual model for sustainable cloud computing is discussed in Appendix F.

7 SUMMARY AND CONCLUSIONS

The use of large numbers of CDCs results in a huge amount of energy consumption and produces significant amounts of large carbon footprints, which has become the greatest challenge of the 21st century. On the other hand, the use of a combination of grids and renewable energy to run CDCs in smart cities can save energy to a large extent. Consequently, there is a need to manage both energy and QoS together to enable sustainable and energy-efficient cloud services. Existing energy-aware resource management techniques and policies mainly focus on VM consolidation to reduce energy consumption of servers only. However, other resources, such as networks, storage, memory, and cooling, consume a huge amount of energy. Efficient scheduling of traffic flow between servers in CDCs is necessary to save energy. Therefore, holistic management of all resources (networks, memory, processors, cooling, and storage) is required to enable sustainable cloud computing. Further, the effect of QoS on the SLA must be addressed in holistic management techniques. Moreover, self-aware or autonomic management of cloud resources in a holistic manner can manage both energy consumption and QoS simultaneously, which can improve the sustainability of cloud computing systems. In addition, dynamically changing the variable clock rates of processors can must optimize energy use. It has also been recommended that the concept *follow the renewable* can motivate cloud providers to locate their CDCs nearer to green energy resources and load can be distributed geographically. However, geographical distribution of resources affects the QoS of networks, which is an open research challenge for the community. Unfortunately, the need to process a huge amount of data and provide high performance simultaneously can also consume large amounts of energy. To solve this problem, energy consumption, SLAs, and QoS must be managed at same time. Further, there is a need for self-aware management of cloud resources holistically to address these research issues. Currently, the research community is working in this direction, but more advanced research is required to ensure the energy efficiency and sustainability of cloud services. In this article, we proposed a taxonomy of sustainable cloud computing to analyze existing techniques for sustainability, including application design, sustainability metrics, capacity planning, energy management, virtualization, thermal-aware scheduling, cooling management, renewable energy, and waste heat utilization for CDCs. Further, the taxonomy mapping-based comparison has been described. A conceptual model for sustainable cloud computing has been proposed. Through a detailed analysis of related studies in the context of taxonomy, we are able to identify and propose various future research directions.

We assert the following conclusions:

- VM consolidation techniques can minimize energy consumption of servers.
- Optimization scheduling of traffic flows between servers is required.
- There is a need for dynamic task scheduling for energy and QoS optimization.
- New system architectures and algorithms can geographically distribute the CDC.
- There is a need for interplay between IoT-enabled cooling systems and the CDC manager.
- Maximum use of renewable energy–powered resources is required for holistic management of resources and workloads.

We hope that this systematic review will be helpful for practitioners and researchers who want to pursue research in any area of sustainable cloud computing.

ELECTRONIC APPENDIX

The electronic appendix (**A:** Background, **B:** Review Methodology, **C:** Elements, Evolution, Comparison, and Open Research Issues, **D:** Outcomes, **E:** Open Challenges and Future Directions: A Summary, **F:** Sustainable Cloud Computing Architecture: A Conceptual Model and **G:** 360-Degree View of Taxonomy) for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

We would like to thank all anonymous reviewers for their valuable comments and suggestions for improving the article. We thank Patricia Arroba, Minxian Xu, and Shashikant Ilager for their useful suggestions.

REFERENCES

[1] Rajkumar Buyya and Sukhpal Singh Gill. 2018. Sustainable Cloud Computing: Foundations and Future Directions. *Business Technology & Digital Transformation Strategies, Cutter Consortium* 21, 6 (2018), 1–10.

[2] Toni Mastelic, Ariel Oleksiak, Holger Claussen, Ivona Brandic, Jean-Marc Pierson, and Athanasios V. Vasilakos. 2015. Cloud computing: Survey on energy efficiency. *ACM Computing Surveys* 47, 2 (2015), 1–33.

[3] Junaid Shuja, Abdullah Gani, Shahaboddin Shamshirband, Raja Wasim Ahmad, and Kashif Bilal. 2016. Sustainable cloud datacenters: a survey of enabling techniques and technologies. *Renewable and Sustainable Energy Reviews* 62 (2016), 195–214.

[4] Sukhpal Singh Gill, Inderveer Chana, Maninder Singh and Rajkumar Buyya. 2018. RADAR: Self-Configuring and Self-Healing in Resource Management for Enhancing Quality of Cloud Services, Concurrency and Computation: Practice and Experience (CCPE), 2018. Retrieved November 24, 2018 from http://buyya.com/papers/RADAR-Cloud-CCPE.pdf. DOI: https://doi.org/10.1002/cpe.4834

[5] Massimo Ficco and Massimiliano Rak. 2016. Economic denial of sustainability mitigation in cloud computing. In *Organizational Innovation and Change*. Springer, Cham, 229–238.

[6] Xiang Li, Xiaohong Jiang, Peter Garraghan, and Zhaohui Wu. 2018. Holistic energy and failure aware workload scheduling in Cloud datacenters. *Future Generation Computer Systems* 78 (2018), 887–900.

[7] Fereydoun Farrahi Moghaddam and Mohamed Cheriat. 2015. Sustainability-aware cloud computing using virtual carbon tax. 2015. arXiv preprint arXiv:1510.05182 (2015).

[8] Josep Subirats and Jordi Guitart. 2015. Assessing and forecasting energy efficiency on Cloud computing platforms. *Future Generation Computer Systems* 45 (2015), 70–94.

[9] Zhou Zhou, Zhi-gang Hu, Tie Song, and Jun-yang Yu. 2015. A novel virtual machine deployment algorithm with energy efficiency in cloud computing. *Journal of Central South University* 22, 3 (2015), 974–983.

[10] Claudio Fiandrino, Dzmitry Kliazovich, Pascal Bouvry, and Albert Zomaya. 2017. Performance and energy efficiency metrics for communication systems of cloud computing datacenters. *IEEE Transactions on Cloud Computing* 5, 4 (2017), 738–750.

[11] Yogesh Sharma, Bahman Javadi, and Weisheng Si. 2015. On the reliability and energy efficiency in cloud computing. In *Proceedings of the 13th Australasian Symposium on Parallel and Distributed Computing*, Parramatta, Sydney, Australia. 111–114.

[12] Dario Pompili, Abolfazl Hajisami, and Tuyen X. Tran. 2016. Elastic resource utilization framework for high capacity and energy efficiency in cloud RAN. *IEEE Communications Magazine* 54, 1 (2016), 26–32.

[13] Dejene Boru, Dzmitry Kliazovich, Fabrizio Granelli, Pascal Bouvry, and Albert Y. Zomaya. 2015. Energy-efficient data replication in cloud computing datacenters. *Cluster Computing* 18, 1 (2015), 385–402.

[14] Muhammad Tayyab Chaudhry, Teck Chaw Ling, Atif Manzoor, Syed Asad Hussain, and Jongwon Kim. 2015. Thermal-aware scheduling in green datacenters. *ACM Computing Surveys (CSUR)* 47, 3 (2015), 1–39.

[15] Konstantinos Domdouzis. 2015. Sustainable cloud computing. In *Green Information Technology: A Sustainable Approach*, Mohammad Dastbaz, Colin Pattinson and Babak Akhgar (Eds.). Elsevier, USA, 95–110.

[16] Zahra Abbasi. 2014. *Sustainable Cloud Computing*. PhD. Dissertation. Arizona State University, Tempe, AZ.

[17] Accenture. 2010. Cloud Computing and Sustainability: The Environmental Benefits of Moving to the Cloud. Online Available at https://download.microsoft.com/download/A/F/F/AFFEB671-FA27-45CF-9373-0655247751CF/Cloud%20Computing%20and%20Sustainability%20-%20Whitepaper%20-%20Nov%202010.pdf.

[18] Prasanna N. L. N. Balasooriya, Santoso Wibowo, and Marilyn Wells. 2016. Green cloud computing and economics of the cloud: Moving towards sustainable future. *GSTF Journal on Computing (JoC)* 5, 1 (2016), 15–20.

[19] Arlitt Martin, Cullen Bash, Sergey Blagodurov, Yuan Chen, Tom Christian, Daniel Gmach, Chris Hyser, et al. 2012. Towards the design and operation of net-zero energy data centers. In *Proceedings of the 13th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm'12)*. IEEE, 552–561.

[20] Francesco Bifulco, Marco Tregua, Cristina Caterina Amitrano, and Anna D'Auria. 2016. ICT and sustainability in smart cities management. *International Journal of Public Sector Management* 29, 2 (2016), 132–147.

[21] Alfonso Capozzoli and Giulio Primiceri. 2016. Cooling systems in data centers: state of art and emerging technologies. *Energy Procedia* 83 (2015), 484–493.

[22] Soundararajan Vijayaraghavan and Joshua Schnee. 2017. Sustainability as a first-class metric for developers and end-users. *ACM SIGOPS Operating Systems Review* 51, 1 (2017), 60–66.

[23] Ana Carolina Riekstin, Bruno Bastos Rodrigues, Kim Khoa Nguyen, Tereza Cristina Melo de Brito Carvalho, Catalin Meirosu, Burkhard Stiller, and Mohamed Cheriet. 2017. A survey on metrics and measurement tools for sustainable distributed cloud networks. *IEEE Communications Surveys & Tutorials* 20, 2 (2017), 1244–1270.

[24] Ryan Bradley, I. S. Jawahir, Niko Murrell, and Julie Whitney. 2017. Parallel design of a product and Internet of Things architecture to minimize the cost of utilizing big data (BD) for sustainable value creation. *Procedia CIRP* 61 (2017), 58–62.

[25] Ruben Van den Bossche, Kurt Vanmechelen, and Jan Broeckhove. 2013. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Generation Computer Systems* 29, 4 (2013), 973–985.

[26] Cinzia Cappiello, Paco Melia, Barbara Pernici, Pierluigi Plebani, and Monica Vitali. 2014. Sustainable choices for cloud applications: A focus on CO_2 emissions. In *Proceedings of the 2nd International Conference on ICT for Sustainability (ICT4S'14)*. 352–358.

[27] Charith Perera and Arkady Zaslavsky. 2014. Improve the sustainability of Internet of Things through trading-based value creation. In *Proceedings of the World Forum on Internet of Things (WF-IoT)*. IEEE, 135–140.

[28] Altino M. Sampaio and Jorge G. Barbosa. 2016. Energy-efficient and SLA-based resource management in cloud data centers. *Advances in Computers, Elsevier* 100 (2016), 103–159.

[29] Charr Jean-Claude, Raphael Couturier, Ahmed Fanfakh, and Arnaud Giersch. 2015. Energy consumption reduction with DVFS for message passing iterative applications on heterogeneous architectures. In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW'15)*. IEEE, 922–931.

[30] Sukhpal Singh Gill and Rajkumar Buyya. 2018. SECURE: Self-protection approach in cloud resource management. *IEEE Cloud Computing* 5, 1 (2018), 60–72.

[31] Ying Zuo, Fei Tao, and A. Y. C. Nee. 2018. An Internet of Things and cloud-based approach for energy consumption evaluation and analysis for a product. *International Journal of Computer Integrated Manufacturing* 31, 4-5 (2018), 337–348.

[32] Sukhpal Singh and Inderveer Chana. 2016. QoS-aware autonomic resource management in cloud computing: a systematic review. *ACM Computing Surveys (CSUR)* 48, 3 (2016), 1–48.

[33] Sukhpal Singh Gill and Rajkumar Buyya. 2018. Failure management for reliable cloud computing: A taxonomy, model and future directions. *IEEE Computing in Science and Engineering* 20, 4 (2018), 1–15.

[34] Li Xiang, Peter Garraghan, Xiaohong Jiang, Zhaohui Wu, and Jie Xu. 2018. Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. *IEEE Transactions on Parallel and Distributed Systems* 29, 6 (2018), 1317–1331.

[35] NoviFlow Inc. 2012. Green SDN: Software Defined Networking in sustainable network solutions. (2012), 1–7. Online Available at https://noviflow.com/resource/green-sdn-software-defined-networking-in-sustainable-network-solutions/.

[36] V. Dinesh Reddy, Brian Setz, G. Subrahmanya, V. R. K. Rao, G. R. Gangadharan, and Marco Aiello. 2017. Metrics for sustainable data centers. *IEEE Transactions on Sustainable Computing* 2, 3 (2017), 290–303.

[37] Hui Zhao, Jing Wang, Feng Liu, Quan Wang, Weizhan Zhang, and Qinghua Zheng. 2018. Power-aware and performance-guaranteed virtual machine placement in the cloud. *IEEE Transactions on Parallel and Distributed Systems* 29, 6 (2018), 1385–1400.

[38] Dirk Pesch, Susan Rea, J. Ignacio Torrens Galdiz, V. Zavrel, J. L. M. Hensen, Diarmuid Grimes, Barry O'Sullivan, et al. 2017. Globally optimised energy-efficient datacenters. In *ICT-Energy Concepts for Energy Efficiency and Sustainability*. Giorgos Fagas, Luca Gammaitoni, and John P. Gallagher (Eds.). IntechOpen, UK.

[39] Min Chen, Yujun Ma, Jeungeun Song, Chin-Feng Lai, and Bin Hu. 2016. Smart clothing: Connecting human with clouds and big data for sustainable health monitoring. *Mobile Networks and Applications* 21, 5 (2016), 825–845.

[40] Sambit Kumar Mishra, Deepak Puthal, Bibhudatta Sahoo, Prem Prakash Jayaraman, Song Jun, Albert Y. Zomaya, and Rajiv Ranjan. 2018. Energy-efficient VM-placement in cloud data center. *Sustainable Computing: Informatics and Systems* (2018). DOI: https://doi.org/10.1016/j.suscom.2018.01.002

[41] Claudia Battistelli, Padraic McKeever, Stephan Gross, Ferdinanda Ponci, and Antonello Monti. 2018. Implementing energy service automation using cloud technologies and public communications networks. In *Sustainable Cloud and Energy Services*. Wilson Rivera (Ed.). Springer. 49–84.

[42] Jong Hyuk Park, Hyun-Woo Kim, and Young-Sik Jeong. 2014. Efficiency sustainability resource visual simulator for clustered desktop virtualization based on cloud infrastructure. *Sustainability* 6, 11 (2014), 8079–8091.

[43] Kai Ding, Pingyu Jiang, and Mei Zheng. 2017. Environmental and economic sustainability-aware resource service scheduling for industrial product service systems. *Journal of Intelligent Manufacturing* 28, 6 (2017), 1303–1316.

[44] Daniel Gmach, Yuan Chen, Amip Shah, Jerry Rolia, Cullen Bash, Tom Christian, and Ratnesh Sharma. 2010. Profiling sustainability of datacenters. In *Proceedings of the IEEE International Symposium on Sustainable Systems and Technology (ISSST'10)*. 1–6.

[45] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering–a systematic literature review. *Information and Software Technology* 51, 1 (2009), 7–15.

[46] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu, and S. U. Khan. 2016. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing* 98, 7 (2016), 751–774.

[47] R. Basmadjian, P. Bouvry, G. D. Costa, L. Gyarmati, D. Kliazovich, S. Lafond, L. Lefèvre, H. D. Meer, J.-M. Pierson, R. Pries, J. Torres, T. A. Trinh, and S. U. Khan. 2015. Green data centers. In *Large-Scale Distributed Systems and Energy Efficiency: A Holistic View*, J.-M. Pierson (Ed.). John Wiley & Sons, Inc, Hoboken, NJ.

[48] Keke Gai, Meikang Qiu, Hui Zhao, and Xiaotong Sun. 2018. Resource management in sustainable cyber-physical systems using heterogeneous cloud computing. *IEEE Transactions on Sustainable Computing* 3, 2 (2018), 60–72.

[49] Cullen Bash, Tahir Cader, Yuan Chen, Daniel Gmach, Richard Kaufman, Dejan Milojicic, Amip Shah, and Puneet Sharma. 2011. Cloud sustainability dashboard, dynamically assessing sustainability of datacenters and clouds. In *Proceedings of the 5th Open Cirrus Summit*. Hewlett Packard, CA. 13.

[50] Tobias Van Damme, Claudio De Persis, and Pietro Tesi. 2018. Optimized thermal-aware job scheduling and control of data centers. *IEEE Transactions on Control Systems Technology* (2018). DOI: https://doi.org/10.1109/TCST.2017.2783366

[51] Dan Azevedo, M. Patterson, J. Pouchet, and R. Tipley. 2010. Carbon usage effectiveness (CUE): a green grid datacenter sustainability metric. In *The Green Grid*. Online Available at http://airatwork.com/wp-content/uploads/The-Green-Grid-White-Paper-32-CUE-Usage-Guidelines.pdf.

[52] Dan Azevedo, Symantec Christian Belady, and J. Pouchet. 2011. Water usage effectiveness (WUE^{TM}): A green grid datacenter sustainability metric. In *The Green Grid*. Online Available at http://tmp2014.airatwork.com/wp-content/uploads/The-Green-Grid-White-Paper-35-WUE-Usage-Guidelines.pdf.

[53] Mark A. Oxley, Eric Jonardi, Sudeep Pasricha, Anthony A. Maciejewski, Howard Jay Siegel, Patrick J. Burns, and Gregory A. Koenig. 2018. Rate-based thermal, power, and co-location aware resource management for heterogeneous data centers. *Journal of Parallel and Distributed Computing* 112 (2018), 126–139.

[54] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam, and Rajkumar Buyya. 2011. Environment-conscious scheduling of HPC applications on distributed cloud-oriented datacenters. *Journal of Parallel and Distributed Computing* 71, 6 (2011), 732–749.

[55] Mung Chiang, Sangtae Ha, I. Chih-Lin, Fulvio Risso, and Tao Zhang. 2017. Clarifying fog computing and networking: 10 questions and answers. *IEEE Communications Magazine* 55, 4 (2017), 18–20.

[56] Sukhpal Singh Gill, Inderveer Chana, and Rajkumar Buyya. 2017. IoT-based agriculture as a cloud and big data service: The beginning of digital india. *Journal of Organizational and End User Computing (JOEUC)* 29, 4 (2017), 1–23.

[57] Anne-Cecile Orgerie, Marcos Dias de Assuncao, and Laurent Lefevre. 2014. A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys* 46, 4 (2014), 1–31.

[58] Monica Vitali and Barbara Pernici. 2014. A survey on energy efficiency in information systems. *International Journal of Cooperative Information Systems* 23, 3 (2014), 1–38.

[59] Praveen Kumar Gupta, B. T. Maharaj, and Reza Malekian. 2017. A novel and secure IoT based cloud centric architecture to perform predictive analysis of users activities in sustainable health centres. *Multimedia Tools and Applications* 76, 18 (2017), 18489–18512.

[60] Sukhpal Singh Gill, Rajkumar Buyya, Inderveer Chana, Maninder Singh, and Ajith Abraham. 2018. BULLET: Particle swarm optimization based scheduling technique for provisioned cloud resources. *Journal of Network and Systems Management* 26, 2 (2018), 361–400.

[61] W. O. Brown Nils, Tove Malmqvist, Wei Bai, and Marco Molinari. 2013. Sustainability assessment of renovation packages for increased energy efficiency for multi-family buildings in Sweden. *Building and Environment* 61 (2013), 140–148.

[62] Chia-Yu Hsu, Chin-Sheng Yang, Liang-Chih Yu, Chi-Fang Lin, Hsiu-Hsen Yao, Duan-Yu Chen, K. Robert Lai, and Pei-Chann Chang. 2015. Development of a cloud-based service framework for energy conservation in a sustainable intelligent transportation system. *International Journal of Production Economics* 164 (2015), 454–461.

[63] Christos N. Markides, 2013. The role of pumped and waste heat technologies in a high-efficiency sustainable energy future for the UK. *Applied Thermal Engineering* 53, 2 (2013), 197–209.

[64] Mueen Uddin and Azizah Abdul Rahman. 2012. Energy efficiency and low carbon enabler green IT framework for datacenters considering green metrics. *Renewable and Sustainable Energy Reviews* 16, 6 (2012), 4078–4094.

[65] Maurizio Giacobbe, Antonio Celesti, Maria Fazio, Massimo Villari, and Antonio Puliafito. 2015. Towards energy management in cloud federation: a survey in the perspective of future sustainable and cost-saving strategies. *Computer Networks* 91 (2015), 438–452.

[66] Anna Kramers, Mattias Höjer, Nina Lövehagen, and Josefin Wangel. 2014. Smart sustainable cities–Exploring ICT solutions for reduced energy use in cities. *Environmental Modelling & Software* 56 (2014), 52–62.

[67] Sukhpal Singh Gill, Inderveer Chana, Maninder Singh, and Rajkumar Buyya. 2017. CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing* (2017), 1–39. DOI: https://doi.org/10.1007/s10586-017-1040-z

[68] Felix Wolf, Bernd Mohr, and Dieter an Mey, eds. 2013. *Proceedings of the 19th International Conference on Parallel Processing (Euro-Par'13).* Vol. 8097. Springer, Aachen, Germany.

[69] Zhao Chen, Ziru Chen, Lin X. Cai, and Yu Cheng. 2017. Energy-throughput tradeoff in sustainable cloud-ran with energy harvesting. arXiv preprint arXiv:1705.02968 (2017).

[70] Ashkan Gholamhosseinian and Ahmad Khalifeh. 2012. *Cloud Computing and Sustainability: Energy Efficiency Aspects*. PhD Dissertation. Halmstad University, Halmstad, Sweden.

[71] Junaid Shuja, Kashif Bilal, Sajjad A. Madani, Mazliza Othman, Rajiv Ranjan, Pavan Balaji, and Samee U. Khan. 2016. Survey of techniques and architectures for designing energy-efficient datacenters. *IEEE Systems Journal* 10, 2 (2016), 507–519.

[72] Chi Xu, Ziyang Zhao, Haiyang Wang, Ryan Shea, and Jiangchuan Liu. 2017. Energy efficiency of cloud virtual machines: From traffic pattern and CPU affinity perspectives. *IEEE Systems Journal* 11, 2 (2017), 835–845.

[73] Maurizio Giacobbe, Antonio Celesti, Maria Fazio, Massimo Villari, and Antonio Puliafito. 2015. A sustainable energy-aware resource management strategy for IoT Cloud federation. In *Proceedings of the IEEE International Symposium on Systems Engineering*. 170–175.

[74] Thomas Dandres, Rejean Samson, Reza Farrahi Moghaddam, Kim Khoa Nguyen, Mohamed Cheriet, and Yves Lemieux. 2016. The green sustainable telco cloud: Minimizing greenhouse gas emissions of server load migrations between distributed datacenters. In *Proceedings of the 12th IEEE International Conference Network and Service Management (CNSM'16)*. 383–387.

[75] Minxian Xu, Amir Vahid Dastjerdi, and Rajkumar Buyya. 2016. Energy efficient scheduling of cloud application components with brownout. *IEEE Transactions on Sustainable Computing* 1, 2 (2016), 40–53.

[76] Tian Wang, Yang Li, Guojun Wang, Jiannong Cao, Md Zakirul Alam Bhuiyan, and Weijia Jia. 2017. Sustainable and efficient data collection from WSNs to cloud. *IEEE Transactions on Sustainable Computing* (2017). DOI: https://doi.org/10.1109/TSUSC.2017.2690301

[77] Sukhpal Singh and Inderveer Chana. 2014. Energy based efficient resource scheduling: a step towards green computing. *International Journal of Energy, Information and Communications* 5, 2 (2014), 35–52.

[78] Jianting Fu, Zhen Zhang, and Dan Lyu. 2018. Research and application of information service platform for agricultural economic cooperation organization based on Hadoop cloud computing platform environment: taking agricultural and fresh products as an example. *Cluster Computing* (2018), 1–12. DOI: https://doi.org/10.1007/s10586-018-2380-z

[79] J. Park and Y. K. Cho. 2018. Use of a mobile BIM application integrated with asset tracking technology over a cloud. In *Proceedings of the 21st International Symposium on Advancement of Construction Management and Real Estate*. 1535–1545.

[80] Saurabh Kumar Garg and Rajkumar Buyya. 2012. Green cloud computing and environmental sustainability. In *Harnessing Green IT: Principles and Practices*, San Murugesan and G. R. Gangadharan (Eds.). Wiley, UK, 315–340.

[81] Sareh Fotuhi Piraghaj, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, and Rajkumar Buyya. 2017. A survey and taxonomy of energy efficient resource management techniques in platform as a service cloud. In *Handbook of Research on End-to-End Cloud Computing Architecture Design*, Jianwen "Wendy" Chen, Yan Zhang, and Ron Gottschalk (Eds.). IGI Global, USA, 410–454.

[82] Abbas Mardani, Ahmad Jusoh, Edmundas Kazimieras Zavadskas, Fausto Cavallaro, and Zainab Khalifah. 2015. Sustainable and renewable energy: An overview of the application of multiple criteria decision-making techniques and approaches. *Sustainability* 7, 10 (2015), 13947–13984.

[83] Sukhpal Singh and Inderveer Chana. 2016. EARTH: Energy-aware autonomic resource scheduling in cloud computing. *Journal of Intelligent & Fuzzy Systems* 30, 3 (2016), 1581–1600.

[84] Mark A. Oxley, Eric Jonardi, Sudeep Pasricha, Anthony A. Maciejewski, Howard Jay Siegel, Patrick J. Burns, and Gregory A. Koenig. 2017. Rate-based thermal, power, and co-location aware resource management for heterogeneous datacenters. *Journal of Parallel and Distributed Computing* 112, 2 (2017), 126–139.

[85] Leandro Cupertino, Georges Da Costa, Ariel Oleksiak, Wojciech Pia, Jean-Marc Pierson, Jaume Salom, Laura Siso, Patricia Stolf, Hongyang Sun, and Thomas Zilio. 2015. Energy-efficient, thermal-aware modeling and simulation of datacenters: the CoolEmAll approach and evaluation results. *Ad Hoc Networks* 25 (2015), 535–553.

[86] Hongyang Sun, Patricia Stolf, Jean-Marc Pierson, and Georges Da Costa. 2014. Energy-efficient and thermal-aware resource management for heterogeneous datacenters. *Sustainable Computing: Informatics and Systems* 4, 4 (2014), 292–306.

[87] Jordi Guitart. 2017. Toward sustainable datacenters: a comprehensive energy management strategy. *Computing* 99, 6 (2017), 597–615.

[88] Xiaoying Wang, Guojing Zhang, Mengqin Yang, and Lei Zhang. 2017. Green-aware virtual machine migration strategy in sustainable cloud computing environments. In *Cloud Computing-Architecture and Applications*, Jaydip Sen (Ed.). InTech, London, UK.

[89] Yuanxiong Guo, Yanmin Gong, Yuguang Fang, Pramod P. Khargonekar, and Xiaojun Geng. 2014. Energy and network aware workload management for sustainable datacenters with thermal storage. *IEEE Transactions on Parallel and Distributed Systems* 25, 8 (2014), 2030–2042.

[90] Hassan Shamalizadeh, Luis Almeida, Shuai Wan, Paulo Amaral, Senbo Fu, and Shashi Prabh. 2013. Optimized thermal-aware workload distribution considering allocation constraints in datacenters. In *Proceedings of the IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. 208–214.

[91] Dong Han and Tao Shu. 2015. Thermal-aware energy-efficient task scheduling for DVFS-enabled datacenters. In *Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC)*. 536–540.

[92] Lijun Fu, Jianxiong Wan, Ting Liu, Xiang Gui, and Ran Zhang. 2017. A temperature-aware resource management algorithm for holistic energy minimization in datacenters. In *Proceedings of the IEEE Workshop on Recent Trends in Telecommunications Research (RTTR'17)*. 1–5.

[93] Hui Dou, Yong Qi, Wei Wei, and Houbing Song. 2017. Carbon-aware electricity cost minimization for sustainable datacenters. *IEEE Transactions on Sustainable Computing* 2, 2 (2017), 211–223.

[94] Sukhpal Singh, Inderveer Chana, Maninder Singh, and Rajkumar Buyya. 2016. SOCCER: Self-optimization of energy-efficient cloud resources. *Cluster Computing* 19, 4 (2016), 1787–1800.

[95] Corentin Dupont. 2016. *Energy Adaptive Infrastructure for Sustainable CDCs*. PhD Dissertation. University of Trento, Trento, Italy.

[96] Patricia Arroba Garcia. 2017. *Proactive Power and Thermal Aware Optimizations for Energy-Efficient Cloud Computing*, Ph.D. Dissertation. Universidad Politecnica de Madrid, Spain.

[97] Marina Zapater, Patricia Arroba, José Luis Ayala Rodrigo, Katzalin Olcoz Herrero, and José Manuel Moya Fernandez. 2015. Energy-aware policies in ubiquitous computing facilities. In *Cloud Computing with e-Science Applications*, Olivier Terzo and Lorenzo Mossucca (Eds.). CRC Press, USA, 267–284.

[98] Ting-Hsuan Chien and Rong-Guey Chang. 2016. A thermal-aware scheduling for multicore architectures. *Journal of Systems Architecture* 62 (2016), 54–62.

[99] Xiaoying Wang, Zhihui Du, Yinong Chen, and Mengqin Yang. 2015. A green-aware virtual machine migration strategy for sustainable datacenter powered by renewable energy. *Simulation Modelling Practice and Theory* 58 (2015), 3–14.

[100] Ranjit Bose and Xin Luo. 2011. Integrative framework for assessing firms' potential to undertake Green IT initiatives via virtualization–A theoretical perspective. *The Journal of Strategic Information Systems* 20, 1 (2011), 38–54.

[101] Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, and Ammar Rayes. 2016. An energy-efficient VM prediction and migration framework for overcommitted clouds. *IEEE Transactions on Cloud Computing* (2016). DOI: https://doi.org/10.1109/TCC.2016.2564403

[102] Maurizio Giacobbe, Antonio Celesti, Maria Fazio, Massimo Villari, and Antonio Puliafito. 2015. An approach to reduce carbon dioxide emissions through virtual machine migrations in a sustainable cloud federation. In *Sustainable Internet and ICT for Sustainability (SustainIT'15)*. IEEE. 1–4.

[103] R. Bolla, R. Bruschi, F. Davoli, C. Lombardo, J. F. Pajo, and O. R. Sanchez. 2017. The dark side of network functions virtualization: A perspective on the technological sustainability. In *Proceedings of the IEEE International Conference on Communications (ICC'17)*. 1–7.

[104] Luftus Sayeed and Sam Gill. 2008. An exploratory study on environmental sustainability and IT use. *Proceedings of AMCIS'08*. 55.

[105] Kateryna Rybina, Abhinandan Patni, and Alexander Schill. 2014. Analysing the migration time of live migration of multiple virtual machines. In *Proceedings of the 4th International Conference on Cloud Computing and Services Science (CLOSER'14)*. 590–597.

[106] Atefeh Khosravi, Adel Nadjaran Toosi, and Rajkumar Buyya. 2017. Online virtual machine migration for renewable energy usage maximization in geographically distributed cloud datacenters. *Concurrency and Computation: Practice and Experience* 29, 18 (2017), 1–13.

[107] Grace Metzger, Alison Stevens, Megan Harmon, and Jeffrey Merhout. 2012. Sustainability opportunities for universities: Cloud computing, virtualization and other recommendations. In *Proceedings of the Eighteenth Americas Conference on Information Systems (AMCIS'12)*.

[108] Sukhpal Singh, Inderveer Chana, and Maninder Singh. 2017. The journey of QoS-Aware autonomic cloud computing. *IT Professional* 19, 2 (2017), 42–49.

[109] Rahul Ghosh, Francesco Longo, Ruofan Xia, Vijay K. Naik, and Kishor S. Trivedi. 2014. Stochastic model driven capacity planning for an infrastructure-as-a-service cloud. *IEEE Transactions on Services Computing* 7, 4 (2014), 667–680.

[110] Yousri Kouki and Thomas Ledoux. 2012. SLA-driven capacity planning for cloud applications. In *Proceedings of the IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom'12)*. 135–140.

[111] Yexi Jiang, Chang-Shing Perng, Tao Li, and Rong N. Chang. 2013. Cloud analytics for capacity planning and instant VM provisioning. *IEEE Transactions on Network and Service Management* 10, 3 (2013), 312–325.

[112] Erica Sousa, Fernando Lins, Eduardo Tavares, Paulo Cunha, and Paulo Maciel. 2015. A modeling approach for cloud infrastructure planning considering dependability and cost requirements. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 4 (2015), 549–558.

[113] Fanxin Kong and Xue Liu. 2016. Greenplanning: 2016. Optimal energy source selection and capacity planning for green datacenters. In *Proceedings of the ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS'16)*. 1–10.

[114] Marcus Carvalho, Daniel A. Menascé, and Francisco Brasileiro. 2017. Capacity planning for IaaS cloud providers offering multiple service classes. *Future Generation Computer Systems* 77 (2017), 97–111.

[115] Daniel A. Menascé and Paul Ngo. 2009. Understanding Cloud Computing: Experimentation and Capacity Planning. In *Proceedings of the International Computer Measurement Group Conference*. 1–11.

[116] Christoph Dorsch and Björn Häckel. 2012. Matching economic efficiency and environmental sustainability: The potential of exchanging excess capacity in cloud service environments. In *Proceedings of the 33rd International Conference on Information Systems (ICIS'12)*. 1–18.

[117] Syed Shabbar Raza, Isam Janajreh, and Chaouki Ghenai. 2014. Sustainability index approach as a selection criteria for energy storage system of an intermittent renewable energy source. *Applied Energy* 136 (2014), 909–920.

[118] F. Pierie, J. Bekkering, R. M. J. Benders, WJ Th van Gemert, and H. C. Moll. 2016. A new approach for measuring the environmental sustainability of renewable energy production systems: Focused on the modelling of green gas production pathways. *Applied Energy* 162 (2016), 131–138.

[119] Adel Nadjaran Toosi, Chenhao Qu, Marcos Dias de Assunção, and Rajkumar Buyya. 2017. Renewable-aware geographical load balancing of web applications for sustainable datacenters. *Journal of Network and Computer Applications* 83 (2017), 155–168.

[120] J. O. Petinrin, and Mohamed Shaaban. 2015. Renewable energy for continuous energy sustainability in Malaysia. *Renewable and Sustainable Energy Reviews* 50 (2015), 967–981.

[121] Eric W. Stein. 2013. A comprehensive multi-criteria model to rank electric energy production technologies. *Renewable and Sustainable Energy Reviews* 22 (2013), 640–654.

[122] Anders S. G. Andrae and Tomas Edler. 2015. On global electricity usage of communication technology: Trends to 2030. *Challenges* 6, 1 (2015), 117–157.

[123] Gang Liu, Ali M. Baniyounes, M. G. Rasul, M. T. O. Amanullah, and Mohammad Masud Kamal Khan. 2013. General sustainability indicator of renewable energy system based on grey relational analysis. *International Journal of Energy Research* 37, 14 (2013), 1928–1936.

[124] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. 2012. Renewable and cooling aware workload management for sustainable datacenters. *ACM SIGMETRICS Performance Evaluation Review* 40, 1 (2012), 175–186.

[125] Sukhpal Singh, Inderveer Chana, and Rajkumar Buyya. 2017. STAR: SLA-aware autonomic management of cloud resources. *IEEE Transactions on Cloud Computing* (2017). DOI: https://doi.org/10.1109/TCC.2017.2648788

[126] Abbas Mardani, Ahmad Jusoh, Edmundas Kazimieras Zavadskas, Fausto Cavallaro, and Zainab Khalifah. 2015. Sustainable and renewable energy: An overview of the application of multiple criteria decision making techniques and approaches. *Sustainability* 7, 10 (2015), 13947–13984.

[127] Xiaomin Xu, Dongxiao Niu, Jinpeng Qiu, Meiqiong Wu, Peng Wang, Wangyue Qian, and Xiang Jin. 2016. Comprehensive evaluation of coordination development for regional power grid and renewable energy power

supply based on improved matter element extension and TOPSIS method for sustainability. *Sustainability* 8, 2 (2016), 143.

[128] Song Hwa Chae, Sang Hun Kim, Sung-Geun Yoon, and Sunwon Park. 2010. Optimization of a waste heat utilization network in an eco-industrial park. *Applied Energy* 87, 6 (2010), 1978–1988.

[129] Kalyan K. Srinivasan, Pedro J. Mago, and Sundar R. Krishnan. 2010. Analysis of exhaust waste heat recovery from a dual fuel low temperature combustion engine using an Organic Rankine Cycle. *Energy* 35, 6 (2010), 2387–2399.

[130] Sotirios Karellas and Konstantinos Braimakis. 2016. Energy–exergy analysis and economic investigation of a cogeneration and trigeneration ORC–VCC hybrid system utilizing biomass fuel and solar power. *Energy Conversion and Management* 107 (2016), 103–113.

[131] James Freeman, Ilaria Guarracino, Soteris A. Kalogirou, and Christos N. Markides. 2017. A small-scale solar organic Rankine cycle combined heat and power system with integrated thermal-energy storage. *Applied Thermal Engineering* 117 (2017), 1543–1554.

[132] Yong Du, Kefeng Cai, Song Chen, Hongxia Wang, Shirley Z. Shen, Richard Donelson, and Tong Lin. 2015. Thermoelectric fabrics: Toward power generating clothing. *Scientific Reports* 5 (2015), 1–6.

[133] Martin Helm, Kilian Hagel, Werner Pfeffer, Stefan Hiebler, and Christian Schweigler. 2014. Solar heating and cooling system with absorption chiller and latent heat storage–a research project summary. *Energy Procedia* 48 (2014), 837–849.

[134] L. M. Ayompe and Aidan Duffy. 2013. Thermal performance analysis of a solar water heating system with heat pipe evacuated tube collector using data from a field trial. *Solar Energy* 90 (2013), 17–28.

[135] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, and Albert Zomaya. A taxonomy and survey of energy-efficient datacenters and cloud computing systems. *Advances in Computers* 82, 2 (2011), 47–111.

[136] Fahimeh Alizadeh Moghaddam, Patricia Lago, and Paola Grosso. 2015. Energy-efficient networking solutions in cloud-based environments: A systematic literature review. *ACM Computing Surveys (CSUR)* 47, 4, 1–32.

[137] Mehiar Dabbagh, Bechir Hamdaoui, Ammar Rayes, and Mohsen Guizani. 2017. Shaving datacenter power demand peaks through energy storage and workload shifting control. *IEEE Transactions on Cloud Computing* (2017). DOI:https://doi.org/10.1109/TCC.2017.2744623

[138] Fredy Juarez, Jorge Ejarque, and Rosa M. Badia. 2018. Dynamic energy-aware scheduling for parallel task-based application in cloud computing. *Future Generation Computer Systems* 78 (2018), 257–271.

[139] Anik Mukherjee, R. P. Sundarraj, and Kaushik Dutta. 2017. Users' time preference based stochastic resource allocation in cloud spot market: cloud provider's perspective. In *Proceedings of the 12th International Conference on Design Science Research in Information Systems and Technology (DESRIST'17)*. 30 May-1 Jun. Karlsruher Institut für Technologie (KIT), Karlsruhe, Germany

[140] Suleiman Onimisi Aliyu, Feng Chen, Ying He, and Hongji Yang. 2017. A Game-theoretic based QoS-Aware capacity management for real-time edgeiot applications. In *Proceedings of the IEEE International Conference on Software Quality, Reliability and Security (QRS)*. 386–397.

[141] D. Kanapram, R. Rapuzzi, G. Lamanna, and M. Repetto. 2017. A framework to correlate power consumption and resource usage in cloud infrastructures. In *Proceedings of the IEEE International Conference on Network Softwarization (NetSoft'17)*. 1–5.

[142] Chao Jin, Bronis R. de Supinski, David Abramson, Heidi Poxon, Luiz DeRose, Minh Ngoc Dinh, Mark Endrei, and Elizabeth R. Jessup. 2016. A survey on software methods to improve the energy efficiency of parallel computing. *The International Journal of High Performance Computing Applications* 31, 6 (2016), 517–549.

[143] Sukhpal Singh, Inderveer Chana, 2013. Consistency verification and quality assurance (CVQA) traceability framework for SaaS. In *Proceedings of the 3rd IEEE International Advance Computing Conference (IACC'13)*. India.

[144] Junaid Shuja, Kashif Bilal, Sajjad Ahmad Madani, and Samee U. Khan. 2014. Data center energy efficient resource scheduling. *Cluster Computing* 17, 4 (2014), 1265–1277.

[145] Junaid Shuja, Raja Wasim Ahmad, Abdullah Gani, Abdelmuttlib Ibrahim Abdalla Ahmed, Aisha Siddiqa, Kashif Nisar, Samee U. Khan, and Albert Y. Zomaya. 2017. Greening emerging IT technologies: techniques and practices. *Journal of Internet Services and Applications* 8, 1, 1–11.

[146] Ignacio Aransay, Marina Zapater, Patricia Arroba, and José M. Moya. 2015. A trust and reputation system for energy optimization in cloud data centers. In *Proceedings of the IEEE 8th International Conference on Cloud Computing (CLOUD'15)*. 138–145.

[147] Eduard Oró, Ricard Allepuz, Ingrid Martorell, and Jaume Salom. 2018. Design and economic analysis of liquid cooled data centres for waste heat recovery: A case study for an indoor swimming pool. *Sustainable Cities and Society* 36 (2018), 185–203.

[148] Atefeh Khosravi and Rajkumar Buyya. 2018. Short-term prediction model to maximize renewable energy usage in cloud data centers. In *Sustainable Cloud and Energy Services*. Springer, Cham, 203–218.

[149] Charalampos P. Triantafyllidis, Rembrandt H. E. M. Koppelaar, Xiaonan Wang, Koen H. van Dam, and Nilay Shah. 2018. An integrated optimization platform for sustainable resource and infrastructure planning. *Environmental Modelling & Software* 101 (2018), 146–168.

[150] Theodore A. Ndukaife and A. G. Agwu Nnanna. 2018. Optimization of water consumption in hybrid evaporative cooling air conditioning systems for data center cooling applications. *Heat Transfer Engineering*, 1–15. DOI: https://doi.org/10.1080/01457632.2018.1436418

[151] Jiahong Wu, Yuan Jin, and Jianguo Yao. 2018. EC 3: Cutting cooling energy consumption through weather-aware geo-scheduling across multiple datacenters. *IEEE Access* 6 (2018), 2028–2038.

[152] Sudipta Sahana, Rajesh Bose, and Debabrata Sarddar. 2018. Server utilization-based smart temperature monitoring system for cloud data center. In *Industry Interactive Innovations in Science, Engineering and Technology*, S. Bhattacharyya, S. Sen, M. Dutta, P. Biswas, and H. Chattopadhyay (Eds.). Springer, Singapore, 309–319.

[153] Morito Matsuoka, Kazuhiro Matsuda, and Hideo Kubo. 2017. Liquid immersion cooling technology with natural convection in data center. In *Proceedings of the IEEE 6th International Conference on Cloud Networking (CloudNet'17)*. 1–7.

[154] Qiang Liu, Yujun Ma, Musaed Alhussein, Yin Zhang, and Limei Peng. 2016. Green data center with IoT sensing and cloud-assisted smart temperature control system. *Computer Networks* 101 (2016), 104–112.

[155] Ioannis Manousakis, Íñigo Goiri, Sriram Sankar, Thu D. Nguyen, and Ricardo Bianchini. 2015. Coolprovision: Underprovisioning datacenter cooling. In *Proceedings of the 6th ACM Symposium on Cloud Computing*. ACM, 356–367.

[156] Sukhpal Singh Gill and Rajkumar Buyya. 2018. Resource provisioning based scheduling framework for execution of heterogeneous and clustered workloads in clouds: From fundamental to autonomic offering. *Journal of Grid Computing* (2018), 1–33. DOI: https://doi.org/10.1007/s10723-017-9424-0

[157] Sathya Chinnathambi, Agilan Santhanam, Jeyarani Rajarathinam, and M. Senthilkumar. 2018. Scheduling and checkpointing optimization algorithm for Byzantine fault tolerance in cloud clusters. *Cluster Computing* (2018), 1–14.

[158] Stelios Sotiriadis, Nik Bessis, and Rajkumar Buyya. 2018. Self-managed virtual machine scheduling in Cloud systems. *Information Sciences* 433–434 (2018), 381–400.

[159] Milad Ranjbari and Javad Akbari Torkestani. 2018. A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers. *Journal of Parallel and Distributed Computing* 113 (2018), 55–62.

[160] Adnan Ashraf and Ivan Porres. 2018. Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system. *International Journal of Parallel, Emergent and Distributed Systems* 33, 1 (2018), 103–120.

[161] Naresh Kumar Reddy Beechu, Vasantha Moodabettu Harishchandra, and Nithin Kumar Yernad Balachandra. 2017. High-performance and energy-efficient fault-tolerance core mapping in NoC. *Sustainable Computing: Informatics and Systems* 16 (2017), 1–10.

[162] C. Dastagiraiah, V. Krishna Reddy, and K. V. Pandurangarao. 2018. Dynamic load balancing environment in cloud computing based on VM ware off-loading. In *Data Engineering and Intelligent Computing*, S. C. Satapathy, V. Bhateja, K. S. Raju, and B. Janakiramaiah (Eds.). Springer, Singapore, 483–492.

[163] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah, and Philippe Merle. 2017. Autonomic vertical elasticity of docker containers with elasticdocker. In *Proceedings of the IEEE 10th International Conference on Cloud Computing (CLOUD'17)*. 472–479.

[164] Yahya Al-Dhuraibi, Faiez Zalila, Nabil Djarallah, and Philippe Merle. 2018. Coordinating vertical elasticity of both containers and virtual machines. In *Proceedings of the 8th International Conference on Cloud Computing and Services (CLOSER'18)*. 1–8.

[165] Sukhpal Singh and Inderveer Chana. 2016. A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing* 14, 2 (2016), 217–264.

[166] Eduardo Felipe Zambom Santana, Ana Paula Chaves, Marco Aurelio Gerosa, Fabio Kon, and Dejan S. Milojicic. 2017. Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *ACM Computing Surveys* 50, 6 (2017), 78.

Received January 2018; revised May 2018; accepted June 2018