Computer Science Review 40 (2021) 100390

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review article

Energy, performance and cost efficient cloud datacentres: A survey

Ayaz Ali Khan, Muhammad Zakarya [*]

Abdul Wali Khan University, Mardan, Pakistan

A R T I C L E   I N F O

Article history:
Received 12 January 2021
Accepted 22 February 2021
Available online xxxx

Keywords:
Clouds
Datacentres
Resource management
Energy efficiency
Performance

ABSTRACT

In major Information Technology (IT) companies such as Google, Rackspace and Amazon Web Services (AWS), virtualization and containerization technologies are usually used to execute customers' workloads and applications — as part of their cloud computing services offering. The computational resources are provided through large-scale datacentres, which consume substantial amount of energy and, consequently, affect our environment with global warming. Cloud datacentres have become a backbone for today's business and economy, which are the fastest-growing electricity consumers, globally. Numerous studies suggest that ∼30% of the US datacentres are comatose and the others are grossly less-utilized, which make it possible to save energy through technologies like virtualization and containerization. These technologies provide support for allocation and consolidation of workloads on appropriate resources. However, consolidation comprises migrations of virtual machines (VMs), containers and/or applications, depending on the underlying virtualization method; that are expensive in terms of energy consumption, performance degradation, and therefore, costs which is mostly not accounted for in many existing models, and, possibly, it could be more energy and performance efficient not to consolidate. This paper describes energy consumption and performance, therefore, cost issues of large-scale datacentres. Besides, we cover various methods for energy and performance efficient distributed systems, clouds and datacentres. We elaborate energy efficiency methods at three different levels: hardware; resource management; and applications. Besides these, different performance management techniques are mapped onto taxonomies and described in details. In last, energy, performance and cost management techniques, at geographically distributed and multi-access edge computing platforms, are described along with critical discussion.

Contents

[*] Corresponding author.
E-mail addresses: ayazali@awkum.edu.pk (A.A. Khan), mohd.zakarya@awkum.edu.pk (M. Zakarya).

https://doi.org/10.1016/j.cosrev.2021.100390
1574-0137/© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Problems such as global warming, national and international energy supply, water complications, growing fuel costs, and computational business economics entirely bring the necessity for energy and performance, consequently, cost-efficient computation into sharp focus. Depletion in power plants that operate using coals, specifically, in the UK, offering an estimated safety margin for energy [i.e. capacity and demand ratio] of just 0.29% in 2017 [1], and the termination of several nuclear power plants in Germany and France, carry the real risk of load-shedding and power outages in the very near future. Due to increase in renewables, a slight increase in the UK energy safety margin can be seen in 2018 (uptake from 29% to 36%). If we presume similar consumption rates to the world of about 3.0% of total energy usage, an approximate 9.6% rise in datacentre energy efficiency will transform to approximately two times growth in the UK's energy safety margin [1,2]. Similarly, [1] also indicates that, until 2020, datacentres energy efficiency will remain unchanged, since industrial private workloads will migrate from internal private clouds to the public clouds. However, due to increase in mobile users and services, Internet of Things (IoT), and computing at scale, an increasing trend in energy consumption of the current datacentres can still be seen. Such an increase in energy consumption and the expected level of service performance would certainly affect the environmental sustainability (3% Greenhouse gases), user's monetary costs and cloud economics [€183.98 billions in 2016 to €217.05 billions in 2017%–18% increase]. For example, Amazon AWS experienced approximately 1% reduction in their sales due to only 100 milliseconds' loss in performance. Therefore, it is essential to look deeply into the problem and identify possible causes, opportunities and appropriate solutions for energy savings and performance improvements (as agreed in Service Level Agreement — SLA document), therefore, cost savings [2,3].

Both economic and ecological problems associated to large-scale, heterogeneous IaaS clouds inspire this research. Due to fast uptake and rise of IaaS private clouds to run academic as well as industrial workloads at the least capital expenses (CapEx), diminishing the operational expenses (OpEx) to power, operate, maintain and cool IaaS resources while ensuring workload performance is a key economical and environmental issue. Assuming their high energy consumption and, subsequent, carbon emissions, it is essential to think for making these IaaS resources more energy, performance, cost (EPC) efficient and ecologically outgoing [4]. This research is conducted with the aim and purpose of gathering current energy, performance and cost-efficient approaches for heterogeneous IaaS clouds to bring a wide-ranging investigation of energy, performance and cost-efficient resource placement, consolidation and management. In IaaS clouds, resource placement and management techniques (e.g. allocation, consolidation, and migration) offer numerous paybacks that could be accomplished over exploiting mechanisms like increasing utilization levels of available resources and diminishing the total number of required resources (hosts) [5,6]. The latter mechanism and methodology also cuts IaaS datacentre's OpEx, cooling costs, carbon emissions and GHGs (greenhouse gases), and lessens electricity bills to increase IaaS provider's revenue and business profits.

In this paper, we debate on the energy consumption of various devices of a system, and offer several taxonomies for energy, performance and cost-efficient resource management mechanisms for large-scale systems covering heterogeneous clusters, and clouds. Important research papers were collected, surveyed and put onto several taxonomies to illustrate, depict and recognize crucial and unresolved topics for more study and examination. We deliberate several state-of-the-art techniques (energy management under workload performance and users' costs limitations), conveyed in the existing cloud literature, that claim improvements in energy, performance and cost efficiencies of large-scale heterogeneous IaaS clouds, and recognize a number of open challenges. We are aware that there is vast amount of existing literature that talks over energy efficiency; however, performance and cost efficiencies are relatively ignored. The major contributions of our work are:

1. we provide a taxonomical overview of state-of-the-art methods in energy, performance and cost-efficient cloud services;
2. we describe various energy management techniques at datacentre level;
3. we ascertain various approaches to performance management;
4. we describe the role of renewables in reducing the ecological and economical impacts of energy consumptions in clouds; and
5. we discuss energy efficiency of hybrid clouds and multi-access edge computing platforms.

The rest of the paper is organized as follows. Section 2 is devoted to cloud, various services and datacentres. The existing energy consumption problem in illustrated in Section 3. Similarly, the existing performance problem is elaborated in Section 4. Section 5 highlights the differences and main contributions of this survey considering the not small amount of already existing surveys about performance and energy in clouds. Various power management techniques, at datacentres level, are discussed in Section 6. An overview of the related work, in the context of performance management methods, is presented in Section 7. Section 8 is devoted to energy efficient systems covering different scheduling techniques at different levels such as systems level, thermal level, network level, renewables that are specific to hybrid, distributed, clouds, and multi-access edge computing, environments, etc. Section 9 is devoted to areas which need further research. Finally, Section 10 concludes this article.

2. Clouds

Large scale systems for instance grids, clusters, and clouds usually consist of a large number of processing nodes connected through networks [7]. These systems carry out a lot of concurrent work as compared to a single system. The non-distributed system such as cluster and supercomputer is used for large mathematical calculations such as weather forecasting, defence and control systems. The distributed systems such as grid and cloud computing are better than non-distributed systems due to reliability, scalability, lower upfront costs and distributed nature of applications (monolithic vs. micro services). Distributed systems are scattered across several distinct geographical locations connected through dedicated and/or third party networks; and, therefore, have high latency services.

Cloud computing provides on-demand delivery of compute resources to users by pay as you go (PAYG) model [8]. The users will only pay for resources they have used during a specific time interval. The computational resources are provided to users according to their demands through the internet. Large scale organizations, consist of hundred or a thousand number of employees, will need hundreds of servers, storage devices and applications. It will require a lot of revenue to buy all these devices and applications. The problem can be solved through connecting hundred or a thousand number of employees to cloud and all the services will be provided through third party service providers. The computational resources consist of hardware (CPU, RAM), software, storage and applications etc. These services can be divided into three different types: (i) Infrastructure as a Service (IaaS); (ii) Platform as a Service (PaaS); and (iii) Software as a Service (SaaS) - as shown in Fig. 1. Besides these, various other terms and notations are used to denote other cloud services such as Container as a Service (CaaS), Agriculture as a Service (AaaS), Security as a Service (SECaaS), Storage as a Service, and many more [8].

2.1. IaaS

The IaaS provides services that are generally located in a datacentre such as servers, storage and networking hardware. These services are usually expensive due to high upfront costs; and, therefore, might not be a feasible option in small businesses. The users will only be billed for the time they have used the infrastructure resources. Moreover, customers are exempted from infrastructure setup costs, as well as, their operational and maintenance cost. Amazon EC2, S3, and Google compute cloud are most widely used and common examples of IaaS. Usually servers are offered in the form of Virtual Machines (VMs), containers and, now, as bare-metal. IaaS providers have usually pre-defined VM sizes that belong to different CPU speeds, memory sizes and even built in software [9]. Largely, IaaS providers sell their resources in the form of VMs and storage. Users are billed, for pre-defined VMs/container types; in currency per hour (using PAYG model —

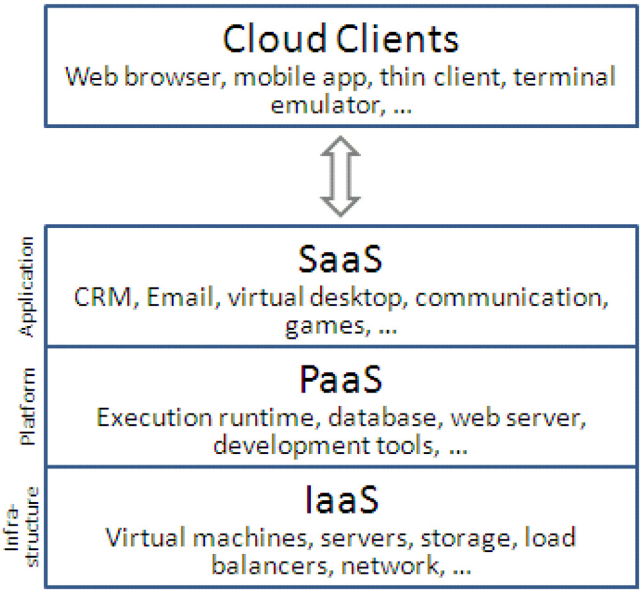
Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure

Fig. 1. The cloud computing stack [IaaS, PaaS and SaaS can be accessed through web browsers, mobile apps and terminals].

pay as you go) based on their workload execution times. AWS EC2, and Google Cloud are common examples of IaaS services. According to Gartner Inc.,[1] the world's IaaS market, while AWS ranked no. 1, grew 21.6% in 2019 to total $38.9 billion, up from $30.5 billion in 2018.

2.2. PaaS

PaaS gives users various computing platforms and services such as web servers, operating systems, computer languages, compilers, and databases etc. It is a framework by which they can design, develop and change certain applications. These services are cost effective for the programmer to develop the software because there is no need to buy the underlying hardware or software. Furthermore, there is no need to spend efforts on preparing and setting up the runtime platform. Google App Engine and online editing, web designing and compilation tools are several examples of PaaS [10]. According to Gartner study, in 2019, the PaaS market values was approximately $19.0 billion which is 17.9% higher than that of 2018. Moreover, the PaaS market is expected to increase by ∼15.3% from 2018 to 2022. As a whole, PaaS services are less popular than IaaS and SaaS; however, the market growth seems inline for all three kinds of services. Prominent PaaS service providers and offerings include AWS Elastic Beanstalk, RedHat Openshift, IBM Bluemix, Windows Azure, and VMware Pivotal CF. Albeit, SaaS market share is currently on top of IaaS; however, as evidenced by the reports from Gartner study, IaaS will have a larger market share and is growing the fastest.

2.3. SaaS

The SaaS provides access to various software which are installed in the remote server (cloud) and accessible to customers through Internet. Usually, these services are offered by third party providers i.e. application providers. These could be e-commerce applications, emails, bank or web services (i.e. web servers). All these cloud services are hosted in large-scale datacentres (IaaS). Furthermore, applications with huge number of users and connections, such as Facebook, customer relationship management (CRM) also run over SaaS environments. These services can be accessed through web browsers [8]. SaaS is a web-based software delivery model in which the software is hosted in a centralized datacentre; and is sold on subscription basis — usually on monthly fee or annual fee. Google App Engine, and on-line games are the most common examples of SaaS service model. The cost of a SaaS application varies with respect to its certain parameters such as the total number of users accessing it. Besides these, certain SaaS providers offer freemium services (with limited functionalities) - Gmail and completely free software. The world's SaaS market was valued $134.44 billion in 2018; and is expected to grow as high as $220.21 billion by 2022.[2]

2.3.1. Datacenters

The datacentre consists of hundred of thousand of servers that can execute multiple tasks/applications at the same time to solve complex scientific and business problems. The virtualization technique is used for increasing system utilization, multitasking and parallelism. It executes multiple Virtual Machines (VM) on a single server that can execute multiple applications. Furthermore, the virtualization technology gives us the opportunity of consolidation by shutting down those servers which are less utilized and, hence, system utilization is increased (workload run over fewer hosts). The datacentre is IaaS in which each service provider may or may not be located on the same geographical location. The example is Amazon EC2 that consists of 22 regions and 69 availability zones and each zone consists of one or more virtualized datacentre.[3] The datacentre consist of hundred or a thousand of nodes and networking devices that consume huge amount of energy/electricity, generate large quantity of heat (CO_2), and require sophisticated cooling systems [11]. The large cooling plants are used to cool these datacentres which will consume a lot of electricity, too. Moreover, storage services are offered from shared and networked storage devices, as shown in Fig. 2.

3. The energy consumption problem

The datacentre consists of hundreds or thousands of nodes so it will consume a large amount of electricity. It will also emit a large quantity of CO_2 emissions and Greenhouse Gases (GHGs) that will pollute the environment. The rise in energy demand and, subsequently, costs by time to time will definitely affect the economics and revenue of service providers. For example, an ordinary server which consume about 450 Wh, its electricity bill is about $352.678 per year at the commercial electricity rate (United States) of $0.08 per KWh. This means that the electricity bill of cluster that consist of up to six thousand servers will cost millions of dollar as energy bill in an year. Therefore, when the energy consumption of the system is decreased then the profit of the service provider will be increased and the GHG emissions will also be minimized. When the GHG is less emitted then the natural environment will be less polluted.

In 2018, datacentre electricity demand was projected to be approximately 198 TWh [11], which is almost 1% of the total electricity demand.[4] Similarly, datacentre networks consumed approximately 260 TWh, which is 1.1% of total electricity demand. Despite the projected 50% increase in datacentre workloads and 80% increase in traffic over the next three years; and due to the current trends for energy efficiency techniques in datacentre

[1] https://www.gartner.com/en/newsroom/

[2] https://www.prnewswire.com/news-releases/

[3] https://aws.amazon.com/about-aws/global-infrastructure/

[4] https://www.iea.org/reports/tracking-buildings/data-centres-and-data-transmission-networks

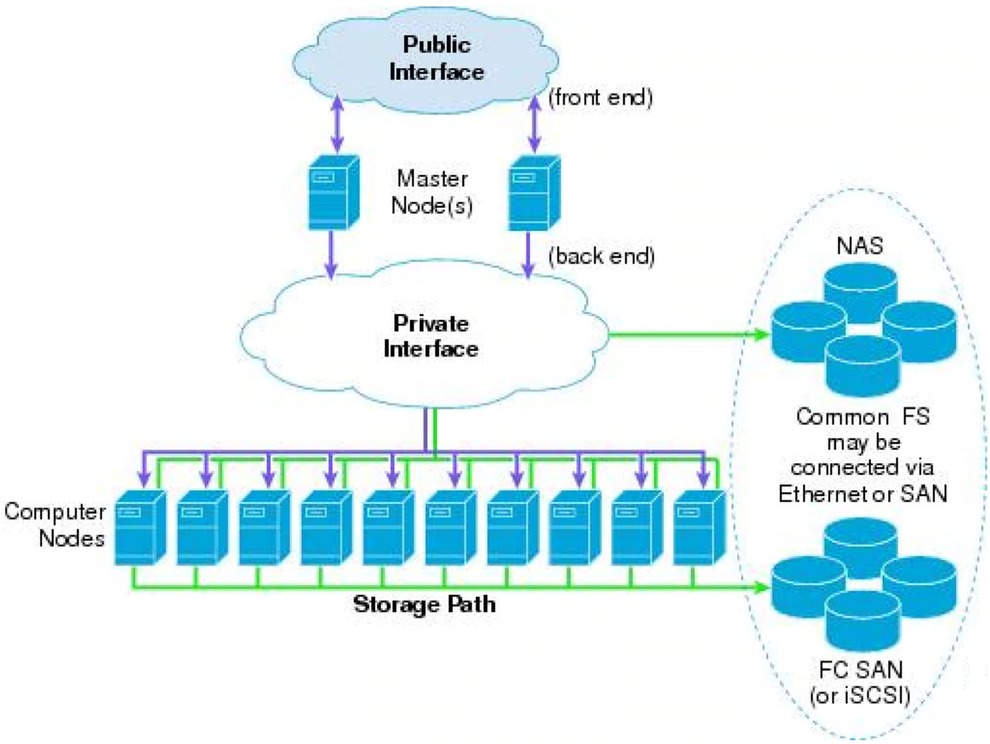

Fig. 2. A view of cloud datacentre — NAS means network area storage [12].

Table 1
Worldwide datacentres energy consumption in TWh [1,13,14].

Year	2015	2016	2017	2018	2019	2020	2021
Servers	88	91	97	102	104	106	109
Storage	14	19	17	18	17	18	19
Infrastructure	84	82	77	74	67	63	59

technologies, their total energy demand is estimated to decrease a bit i.e. 191 TWh in 2021 [1,13] – as shown in Table 1. According to [1,13], the energy sector is responsible to emit approximately 43% of the GHGs. The traditional computing is converted into green computing by minimizing the GHG emissions. The GHG emissions is minimized by minimizing the energy consumption of large scale systems.

If we, now, assume moderate improvements in energy efficiency i.e. 10% per year [11], the current demand for electricity could rise up to 10% i.e. approximately 280 TWh by 2021. Furthermore, assuming a 20% energy efficiency improvements per year, the electricity demand may drop up to one fourth i.e. 25%; therefore, leading to 190 TWh in total. Keeping in view the possible growth of online services, games, mobile devices, and cloud computing, the main aim of service provider will be to look for techniques to reduce the energy consumption, increase their revenues, being environment friendly, in such a way that performance of these systems is not negatively affected.

As discussed before, being efficient at all cost may be even counter-productive both economically and ecologically; particularly when the power supply is varying due to renewable energy sources. Furthermore, apparently energy efficient hardware may mean that certain workloads need to run for longer, and the trade-off between efficiency and runtime, as well as the implication in cost of increased runtime, all need to be addressed. The current trends of cloud service providers towards using renewable energy sources that may operate intermittently, and hence necessitate falling back to the energy grid, also implies a need for consolidation policies to be able to effectively switch between the available energy sources [15], as well as to reduce the replacement cycle of renewable capture and storage equipment. With 640 datacentre outages in the UK alone in 2015 and outages expected to be more common in near future,[5] there is a need at least for proper capacity planning, consolidation of workloads onto servers powered by renewables, and migration of workloads when it is most energy, and therefore cost, efficient, to safeguard supply and reduce the drain on renewable generation and storage equipment. However, greater energy efficiency may mean decreased performance at the same price given dependencies of various workloads on CPU architectures [16]. Ideally, a general utility computing model would emerge as a cloud-based service to provide the best trade-off between performance, cost and energy requirements.

4. The performance issue

Growth in online services (4.2 trillion gigabytes per year), games and, in particular, mobile phones (3.6 billion in 2018 to 5 billion by 2025), IoT devices (7.5 billion in 2018 to over 25 billion by 2025) along with real-time applications tends towards exponential growth in demand for datacentre and networked clouds services [13]. As, these devices generate huge amount of data which is not possible to process, locally. Therefore, these applications suffer from severe performance degradation due to higher latencies of networks. Furthermore, cloud users are billed for how longer they provisioned resources for their workloads. Therefore, if performance requirements of their workloads are not met; it will cost them for more money and may result in SLA violations that could subsequently affect providers revenue and quality of services.

Cloud applications suffer from severe performance degradation due to several reasons. For example, virtualization, and containerization have extra burden of an additional layer over the

[5] http://www.greendatacenternews.org/articles/share/887707/

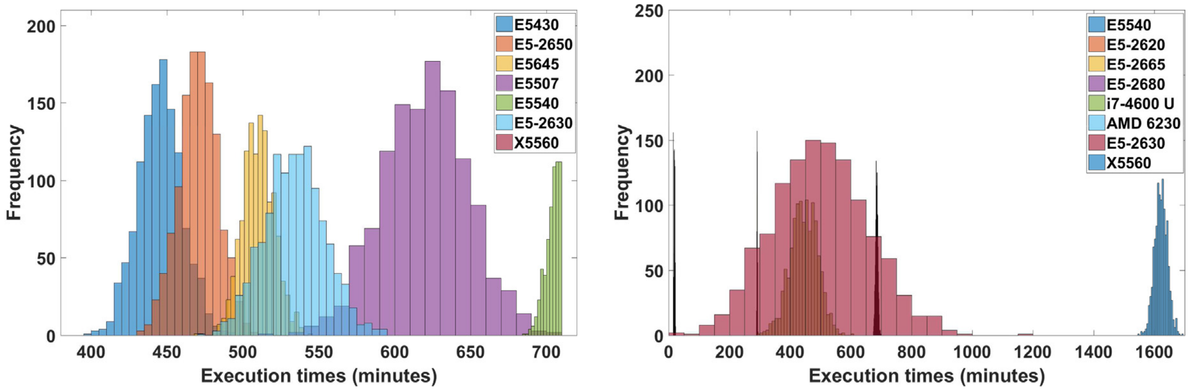

Fig. 3. Variations in applications performance when running over various CPU models performance of the Bzip2 (right) and Povray (left) workloads significantly varies across various platforms.

bare-metal infrastructure (hypervisor). Moreover, VMs and containers could create further issues if they are co-located on similar machines and compete for same resources. Various organization, such as SPEC,[6] and researchers have benchmarked the cloud infrastructure, application (SaaS) and resources performance [17]. Performance degradation of virtualized and containerized systems is further explained in [18]. In real-time scenarios, such as autonomous cars, performance will essentially matter; and if not then customers will switch to other providers.

Therefore, with the maturity of public clouds and services, their capabilities to manage workload performance should be improved. Furthermore, in order to provide better performance, service providers should gradually learn and evolve their offerings in terms of instances, sizes and types, resource management policies i.e. resource allocation, workload scheduling and consolidation with migration, and resources (CPU architecture, graphic cards, GPUs) [19]. We are aware that prices of workloads and instances are set in SLA's and improvements in their performance might not be desirable at all. However, there are certain applications, such as database queries and high performance systems, where user satisfaction could be achieved through higher performance gains. Moreover, users monetary costs, and energy efficiency of the infrastructure should be associated to performance of the provisioned resources. A possible reason for this performance variations is probably the CPU architecture and platform. For example, as shown in Fig. 3, the performance of various workloads vary significantly across various CPU models. Besides increasing users' costs, user might be able to choose the right instance/CPU model that may affect the IaaS providers resource utilization and revenues. This phenomena is known as instance seeking in the cloud literature [17].

The existing trade-off between energy consumption and workload performance makes the resource management a non-trivial activity. Previous surveys [2,9,20–22], have largely discussed energy consumption of datacentres; however, the performance management techniques and their impact on energy efficiency and environmental sustainability is relatively unexplored. This study has been completed with the intentions to provide a comprehensive review of energy consumption and performance of application along with their impacts on cloud economics, and ecological concerns. Furthermore, we offer state-of-the-art methods in emerging computational services like multi-access edge computing, and hybrid datacentres.

5. Motivation

Both economical and environmental issues related to large scale datacentres motivate us for this study. With the rapid uptake of cloud datacentres to host industrial applications, reducing the operational costs of powering and cooling large scale datacentres, such that the workload performance is not affected, is a major economical concern. As for every 10 °C increase in temperature, the system failure rate doubles, hence reduced temperature could also improve datacentres reliability [23]. Various studies are conducted to elaborate and investigate green computing and datacentres. The objective of this survey is to analyse the energy consumption of ICT devices in general (including Laptops, PCs etc.) and focus on large scale HPC systems, clusters, grids, clouds and particularly datacentres. We explain a taxonomy of techniques that are proposed to enhance the energy and performance efficiency of these systems. Conceptually clusters, grids and clouds are treated the same [24], hence considerable efforts have been made to analyse and differentiate the energy efficiency methods proposed for these systems. The major contributions of the survey are as follows:

1. a taxonomy of power and performance efficient cloud computing;
2. a review of system level energy efficient CPU scheduling in single systems along with different types of schedulers;
3. discussion of various cluster level scheduling techniques to diminish the energy consumption of datacentres under the performance constraints; and
4. taxonomy of datacentre level resource management techniques in terms of energy and performance efficiencies for virtualized and containerized clouds.

Our survey is different from those conducted in [2,9,20,25,26]. In this survey, we extend our own survey conducted in [20] in order to account for: (i) performance aware computing; (ii) energy-performance efficient datacentres; and (iii) new findings and directions for future research. The techniques presented in [9], provide a taxonomy of optimizations, but the energy efficiency techniques in virtualized cloud environments have not been studied. Similarly, the surveys conducted in [2,25–27] only focus on energy efficient datacentres and have ignored the energy efficiency of system level CPU and resource scheduling. These studies have also ignored the performance efficiency of compute clusters that form a base for grids and cloud systems. Furthermore, performance of cloud's workload is not explored. We start from the energy and performance efficiency of a single system (its different components) and explore the energy and performance efficiency of large scale cloud datacentres, storage systems and

[6] https://www.spec.org/cloud_iaas2018/

networking. We believe that this survey will help readers to understand the necessary concepts of energy-performance-cost efficient resource management techniques (to achieve energy efficiency) in cloud systems including bare-metal, virtualization, containerization and a mix of all technologies. Furthermore, this survey will help readers to highlight the key and outstanding issues in energy-performance-cost aware datacentres for further research.

6. Power management techniques

The methods which are used to optimize the energy efficiency of computer systems are known as power management techniques; that could be achieved in three different ways: (i) design the device in such a way that it needs less power to operate i.e. Static Power Management (SPM); (ii) operate the device at low power if feasible or simply switch it off when it is not in use i.e. Dynamic Power Management (DPM); and (iii) application level approaches that make use of energy efficient compilers, software development, and programming [9,28]. In respect of (ii), there are two different methods: (a) hardware-based capabilities which are embedded in certain devices such as DVFS, ALR; and (b) software-based methods or algorithms that reduce energy consumption through adaptation of the system behaviour based on the resource demand [29–31]. These three techniques are, then, placed to a taxonomy, for classification, as shown in Fig. 4.

6.1. Static Power Management (SPM)

Static power management techniques keep an idle system or device in power efficient state until it is more utilized [9]. These techniques are suggested to be efficient on a single system, but, not distributed systems. It consists of several states from stand by state to power off state. The power consumption of a less utilized system is further reduced as compared to more utilized systems. It is applied to different server components such as processor, disk, chip, communication link, etc. SPM techniques include reducing the number of power switches in logic gates from one state to the other, clock and power gating, and static leakage management (SLM). The clock gating concept is used to reduce the power consumption of the processor. A clock gate consumes about 60%–70% of total chip energy. The chip power is directly proportional to capacitance C, Voltage V, Switching activity A and frequency f [32]. SPM methods are mainly divided into two different levels. The first one is a low level approach at CPU level. It investigates the power consumption of a CPU at instruction and cycle level. The second approach handles the power consumption of all the server components.

6.2. Dynamic Power Management (DPM)

Dynamic power management methods reduce the power consumption of the system without degrading its performance. It will change the system behaviour according to resource demands and current usage [12,29]. The systems that are idle are shifted into sleep mode and wake them when it is in working state. Note that, DPM is the application level management of the system. Moreover, it is better than SPM in single or large systems. These techniques can be divided into two types: (a) hardware-based; and (b) software-based.

6.2.1. (a) Hardware-based DPM

In hardware-based, certain capabilities of the hardware can be used to adapt the hardware energy consumption as needed by the resource demand. For example, DVFS [32] and ALR [33] techniques can be used if the CPU has noting to carry out or the network has no packets to transfer, respectively. Other hardware-based techniques are Dynamic Voltage Scaling (*DVS*), Dynamic Power Scaling *DPS* etc. The hardware-based technique is dependent on the server hardware like the DVFS enabled processor which will scale up or down the speed of the processor according to the system utilization.

6.2.2. DVFS

The Dynamically Voltage Frequency Scaling (*DVFS*) dynamically adjust the speed and power of the system to minimizes the energy consumption of the system [32]. The energy consumption of the system is minimized but the performance of the system is degraded due to the execution of task on slower speed. The DVFS can be used by its optimal by slowing the speed of the processor. Energy saving methods, like Dynamic voltage scaling (DVS), help in reducing the energy by achieving dissipation for the core by steadily lowering the voltage as well as operating frequency. At the moment, more efforts are implied to research and development for the DVS methods. Furthermore, DVS approach constitutes to standardize the balance amongst performance and battery life keeping in the context of two major properties. They are: (i) sustenance must be achieved to have average throughput less than that needed for high computing power; and (ii) the CMOS base logic is used in processors [9].

The prior property reflects the need of high performance is required for a small span of time, whereas for the rest of time, low performance and low powered processor is enough. Low performance is achieved by reducing operating frequency for the processor whenever peak speed is not further required. As a resultant, DVS balances processor's operating voltage as well as frequency. As a standard, DVS is employed to manage power usage of the system. Based on the truth, the dynamic power (switching) P for CMOS circuit/s is fully dependent upon core voltage V, and the clock frequency f accordingly. As it is evident that execution time has inverse proportionality with frequency, therefore for computation the total energy E is proportional to square of the voltage.

The work presented in [22] discusses methods to enhance energy efficiency for computation and network resources being part of large-scale distributed systems using dynamic voltage and frequency scaling (DVFS), or by consolidation (cluster VMs to elude power to many hosts). In terms of compute resource, the method works at several levels i-e, from single node to whole infrastructure through which an advantage may be taken in the form of virtualization. But the reduced energy usage yields limited affect as it decreases performance. This also leads to have a low expected gain for reduction in energy efficiency; as users are directed to use further compute resource if are cheaper in cost. Therefore, it is not a proper solution for rapidly growing large-scale datacentres where compute power is considered keeping reduced carbon emissions.

6.2.3. ALR

In 2018, both fixed and mobile networks were responsible to consume approximately 261 TWh of energy which is equal to 1.1% of the total global electricity demand [11]. Various techniques can be used to reduce the energy consumption of networks and related devices. For example, the adaptive link rate (ALR) technique is based on the mechanism that when the network is less utilized or there are no packets to transfer; then, slows down the speed of network [21,33]. There is a negligible

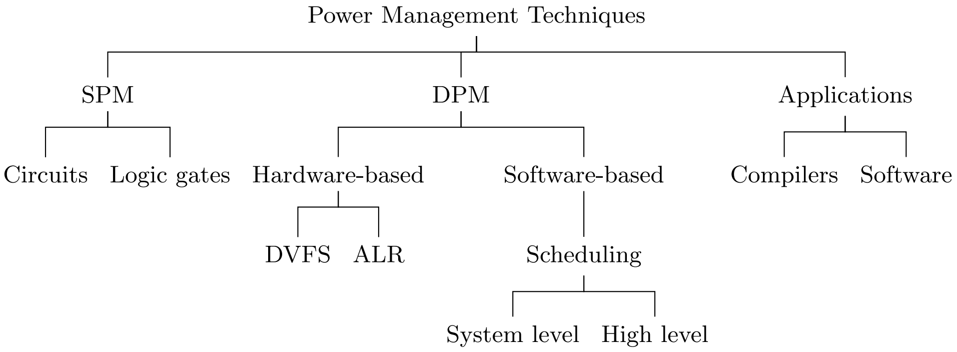

Fig. 4. Taxonomy of power management techniques.

difference between less or fully utilized network so the energy difference by using ALR technique will also be not significant. It uses queues to store the incoming packets when the network or receiver system is in idle mode. Moreover, it will result in delaying the data transfer. When the system changes its state from idle to running then all these packets are transmitted which will result in congestion. The congestion will increase transmission time and energy consumption of the network. Various congestion control mechanisms can be used such that the congestion is avoided [12]. For example, the energy efficient Transmission Control Protocol (TCP) is used to avoid congestion as compared to simple TCP in two different ways: (i) the acknowledgement message from the receiver to the sender does not have enough information about the state of the receiver; (ii) when there are bursts of errors from sender to receiver it will inform the TCP about these errors [34]. In respect to (i), the packets lost or out of order — it will use the acknowledgement mechanism to retransmit it. In respect of (ii), if there are burst error then the TCP will burst it; otherwise send an acknowledgement to sender. By using the energy efficient TCP about 75% energy can be saved [2].

6.2.4. (b) Software-based DPM

Software-based techniques use various resource management methods and policies to: (i) activate hardware-based techniques to achieve power efficiency in large distributed systems like clusters, grids and clouds; or/and (ii) decide energy efficient workload placement and migration on appropriate servers. The software-based technique like the uniprocessor scheduling when applied to hardware-based approach like the DVFS, large energy savings could be achieved. These are the hardware capabilities to adapt their energy consumption and performance according to the workload demand fluctuations. For example, ALR (adaptive link rate) is designed to slow the data transfer rate across a network (connections) when feasible.

6.2.5. System level scheduling

The software-based DPM can be achieved through system level scheduling. The system level scheduling will give tasks to the processor in such a way that all the tasks are executed in less time or on energy efficient processor. So the scheduling can be used to arrange the tasks in such a way that the overall execution time of the tasks are reduced. Furthermore, the power consumption of the system is reduced by reducing the overall execution time. There are three types of system level scheduling which are: (a) uniprocessor scheduling; (b) multiprocessor scheduling; and (c) multicore scheduling [32,35,36]. The uniprocessor consists of only one processor and the tasks are simply allocated to only one processor i.e the Earliest Deadline First Algorithm (EDF), Rate Monotonic Algorithm (RM) and Total Bandwidth Server (TBS) etc. The multiprocessor scheduling is complex as compared to uniprocessor, because the task is optimally allocated to only one processor out of n multiprocessors. In multicore, tasks are allocated to the best core out of m multiple cores. These three scheduling techniques are deeply discussed Section 8.

6.2.6. Higher level scheduling — resource management

Resource management in large scale systems such as cluster, grid and cloud is critical. It efficiently use the resources of large systems and provides the Quality of Service (QOS) to end users. The cluster and grid system provide the best effort services to users. Whereas, the cloud provides reasonable resources to customers [12]. These large scale systems use resource management techniques to solve complex mathematical calculation, strict delay systems, and service delivery systems. If these systems does not deliver the required performance, then, the user will not pay for it. The system will deliver the required performance through energy and performance aware resource management, scheduling and consolidation techniques. Various techniques, which are used in the literature, to reduce the energy consumption of clusters and datacentres are described in existing works [3,18]. Usually, clouds are distributed over various geographic areas, which have different energy sources such as grid energy, renewables, solar and various prices. Furthermore, providers would take economical benefits from renewables rather than grid energy; however, users workloads might be suffered and, therefore, the costs they pay for their provisioned resources. Furthermore, the performance of resources are also subject to the quality of the underlying network. These diverse options should be accounted when service providers deal with their customers in terms of resource allocation, consolidation and management decisions.

6.3. Applications

Apart from SPM and DPM, energy efficiency can also be achieved at application level [3,9,29]. The basic idea is to design software code in such a way that the developed application could run using less power (e.g. using fewer instructions). Moreover, compilers can be developed to produce fewer instruction and are enough intelligent to speculate when the processor can be switched to idle state. Note that, application level energy efficiency and performance management techniques are not within the scope of this research. Apart from these methods, researchers have also proposed self-configuration and adaptation (intra-layer i.e. SaaS, PaaS, IaaS) based approaches to increase energy efficiency across the entire cloud stack [2]. Numerous methods like application profiling (SaaS), manager (PaaS) and VM monitoring, sizing, and scaling tools (IaaS) are used to interconnect these layers in order to take appropriate decisions for energy efficiency.

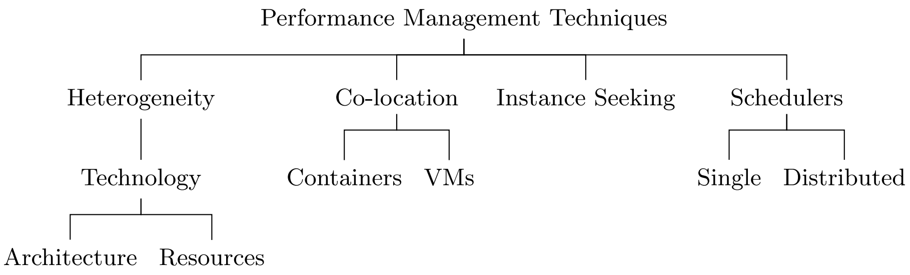

Fig. 5. Taxonomy of performance management techniques.

7. Performance management techniques

In this section, we describe various methods which are used to manage computational performance of cloud services and applications. Performance of applications varies, few applications would prefer small latencies while some may account for execution times. For example, real-time and deadline-oriented services may need quick response time, while, certain database services and batch workloads would need to be executed enough fast to reach a solution. Hence, workload execution times suffer users' costs; therefore, we may recall performance as application runtimes onwards, in this paper. Known methods, policies and offerings are mapped to a taxonomy, as shown in Fig. 5, and described in subsequent sections.

7.1. Technological growth

Recent advanced technologies can be used to improve datacentres performance. For example, GPU servers and Intel Optane persistent memory (which sits between DRAM and SSDs)[7] can be used to run CPU and data intensive workloads, respectively. Edge datacentres is an approach to move execution from centralized stations closer to users in order to reduce processing delays. Similarly, advancement in machine learning, artificial intelligence and high performance computing (HPC) systems (parallelism and distributed computation) will help to improve cloud performance. Other methods including containerization, Functions as a Service (FaaS), micro-services, and server-less architecture have enabled vast adaptation of cloud services for performance benefits. Certain applications would perform better in containers, while other in VMs that will transform current datacentres into hybrid architecture. Besides these, the exponential growth in microprocessor technology, relating Moore's law, has significantly improved performance of general-purpose processors which are also deployed in today's datacentres [37].

7.2. Heterogeneity

Heterogeneity is the term used to denote architectural differences in various hardware that affect application performance [38]. Literature suggests that certain application may perform worse on a particular CPU but the same CPU model would be better for another application. This heterogeneity can be used to improve application performance through resource management techniques. For example, if an application is not performing up to the expected level on a machine; then, it should be migrated to another best performing machine [39]. For example, as shown in Fig. 3, the performance of various applications vary significantly on different CPU models. It is possible that a certain instance or workload would perform quite differently on two same or different CPU architectures [17]. Cloud users could take additional performance benefits while launching appropriate instance type for their particular workload; if, such predictability of the platform's performance can be realized in advance. The known phenomena is call instance seeking [40]. Largely, the distribution of workload runtimes follows a log-normal pattern across different CPU models; and this may happen due to either CPU heterogeneity and/or resource contention. Moreover, a particular workload may run quickly on a specific CPU model, but, may run quite slow on another CPU model. Similarly, a CPU model may run a particular workload quickly, but, another one quite slow. For example, E5430 is faster for bzip2 benchmark than E5507, but, is slower for povray benchmark — as shown in Table 2.

Table 2
Execution times (seconds) of various applications across different CPU models [16].

Workload type	CPU model	Execution times
bzip2	E5430 E5507	447 s 641 s
povray	E5430 E5507	579 s 544 s

7.3. Co-location

Datacentre co-location refers to cloud resources from various vendors and providers, but, co-located in one large-scale datacentre [17]. This may also refer to cloud services from one company but offered at different geographical locations — edge cloud. The reason is that companies will put their resources where they are most needed and are economically rich. Furthermore, certain VMs and containers that run a single user application can be placed co-located on same machines [41,42]. However, these should not be confused as the latter one is a different technique than the former ones. As described above, regarding resource contention several co-located VMs on a specific host may experience severe performance degradation, particularly, if they compete for same resources (resource interference). The degradation is dependent on the total number of co-located VMs and the workload type they are running on a particular host — as shown in Table 3. The more number of VMs are co-located on a single host, the more performance degradation can be observed for running applications. Moreover, these performance variations vary with respect to workload types.

7.4. Instance seeking

A large number of instances are being provided by cloud service providers — every instance has some unique resources like vCPU, storage, memory and network status. Some cloud instances are optimized for heavy computation like CPU intensive workloads, memory, AI, and many more. This enables the choice of picking the right instance for the right workload or

[7] https://www.datacenterknowledge.com/hardware/intel-s-new-memory-technology-boosts-data-center-performance

Table 3
Execution times (seconds) of various applications on co-located VMs [41].

Workload type	CPU model	Number of co-located VMs					
		2	4	6	8	10	12
		Execution times					
Grep	E5620	13	14	16	21	31	36
	E7420	20	22	25	29	38	44
Sort	E5620	16	22	38	59	69	78
	E7420	21	28	43	65	76	85

application [40]. If the size of an instance is too large, then additional resources will have low benefit over cloud performance (workload) eventually wasting money. On the other hand, if the size of an instance is too small then it will have diminishing effect on performance — the case of workload running at all. Few works [17,40], have demonstrated the predictable performance of various instances, offered in different availability zones on the AWS cloud, that can be related onto original bare-metal hardware which host them. For example, an instance created in one zone is hosted on a particular CPU model; while similar instances are placed onto different hardware in another zone. These predictable performance would enable users to choose a particular zone while launching their instances to take benefits of the performance. The problem of launching preferable instances of own choices is known as instance seeking; which might have negative impacts on providers revenue and resource management. For example, as shown in [43], instance seeking has a negative impact of the resource allocation, therefore, energy efficiency and performance.

7.5. Micro-service architecture

Large applications encloses the significant characteristics along with functionalities into a single executable structure. Though the software development method is *"tried and true"* but such applications present scalability and performance challenges in cloud scenario [44]. As soon as a performance limit is reached for conventional large applications, a brand-new deployment of the instance is done. Applications are broken up into series for inter-related individual programs, scaled and operated through micro-services. The divided independent services are managed to work with each other through some APIs hence providing characteristics and functionality for such applications. Once, a service is heavily loaded to its performance then that service is scaled out. Micro-service architecture enables the development of a software in independent pieces rather than a single suit. This makes its delivery faster; but, faster development and deployment does not necessarily suggest good performance of applications [45]. In order to ensure the best possible performance of applications deployed using micro-service architecture, it is essential to: (i) optimize architecture; (ii) code; (iii) wise deployment of containers; and (v) effectively monitor the whole infrastructure. Load balancing, auto-scaling and appropriate resource management are various methods which have shown performance improvements in micro-services.[8]

Micro-service architecture is designed to offer agility, and scalability during the software development process. However, due to large number of micro-services, application performance prediction is difficult. Several works have suggested methods to diagnose the application performance. For example, [46] proposes MicroRCA, a system to identify major causes for performance issues in micro-services. This is achieved through correlating performance symptoms of the application with resource utilization levels of the corresponding system. Their evaluations, in a Kubernetes cluster, demonstrate the precision and accuracy of the proposed system. As described later in Section 7.8, performance of the micro-service applications can be improved through auto-scaling techniques.

7.6. Cache

The copy of frequent data accessed is placed at the fastest available storage called as cache which is situated very close with executing application [17]. Moreover, cache can also be integrated into application. Very similar to the concept of cache in a single system, the availability of data at cloud cache storage enables faster execution of the data as compared with transfer of data from regular storage. Largely, the amount of data to be transferred very frequently can be reduced, along with other benefits such as reduced network traffic. Different cache services are available like Microsoft Azure cache for Redis, ELasticache[9] of Amzaon and Memcache for Google App Engine. The problem associated with cache storage is that if there is some change in original data then that change may not be updated in cache as it has copy of previous original data. This causes problems in case of using cache storage. Since, cache data can be accessed faster than storage data, therefore, application performance can be boosted. This could also help in predictable performance which is a common challenge in modern applications to scale their resources accordingly.

Cache is usually used to enhance database performance and data transmission. The AWS Elasticache runs like a service and offer a cluster of caches. There are other ways to cache the application's data, e.g. store each application's data on the local host where the application resides. Largely, the cache can boost the performance of web applications. Using a cache can reduce the costs of databases and load on the backend server. Leveraging cache appropriately could result in an application that not only performs better, but also costs less at scale [47]. Applications which involve heavy I/O (input/output) activities, that can be seen very common in the cloud, may benefit the most from cache as a service (CaaS). In [48], the authors suggest that CaaS offer benefits both to customers and service providers i.e. performance and profit, respectively. The CaaS model leverages the cloud economy in that: (i) the additional customer costs for I/O performance gains are minimal; and (ii) the provider's profit grows due to larger opportunities in server consolidation that result from performance gains. Furthermore, [49] suggests that a marginal increase in cache hit rate of 1% can decrease the latency of application layer by more than 25%. Moreover, they propose a cache controller i.e. "Dynacache" which significantly improves the hit rate of web applications. However, these variations in performance gain are dependent on workload types.

7.7. Server-less architecture

The developers are always attracted by server-less based computing which suggest that the cloud user simply writes the code and leaves all the resource provisioning, maintenance, and administration tasks to the cloud service provider [50]. It enables event driven frameworks to run on server-less services within the cloud like Google cloud function, Microsoft Azure function and AWS lambda. The notions of FaaS (function as a service) and BaaS (backend as a service) usually refer to server-less computing in well-known public clouds. Though event driven functions

[8] https://www.oreilly.com/library/view/production-ready-microservices/9781491965962/

[9] https://aws.amazon.com/elasticache/

are operated by servers at the back end, the main objective is to bypass deployment as well as long-standing operation of conventional instances of VM/containers. As an alternative, large scale developers uses software behaviours/functions to load code onto the cloud structure. Here, when a real world or automated event is triggered then the code is deployed and executed. On completion of the function, it is unloaded to release any further usage of cloud resource. Subsequently, through usage of server-less component, the size of overall application is small, simple and optimally optimized for performance [44,51].

Deploying micro-service applications onto server-less platforms result in cost savings as compared to traditional applications. In [52], authors have described implementation challenges of the server-less platform, for example, function scaling, container discovery, and reuse. Moreover, various metrics are suggested to evaluate the performance of server-less platforms such as AWS Lambda, Azure Functions, and Google Cloud Functions. In [53], authors have presented a comprehensive investigation on various factors that affect the performance of micro-service applications hosted on server-less platforms. The study considers scalability, load-balancing, and variations in resource provisioning. Server-less computing platforms offer more flexibility, quick deployment, greater scalability, and shorter periods to release resources, at a reduced cost. Moreover, developers are not worrying about purchasing, provisioning, scaling, and managing the backend servers. Moreover, code can run closer to the end user which reduces application latencies. However, server-less platforms are not recommended for log-running services and can increase user costs. Performance may vary in terms of cold start and warm start. A cold start occurs when a function in not ready for execution and needs to be loaded before its execution. Similarly, a warm start occurs when a function is ready to execute. The former one would certainly affect the application performance. Further details on performance of application hosted in server-less platform can be found in [[10]].

7.8. Auto-scaling services

The typical nature of public cloud infrastructure is dynamic. It has the capability to add/remove instances and related resources on arrival of demand. Auto-scaling and load balancing is offered by Microsoft Azure, AWS and Google Cloud Platform (GCP). A suitable rule set implemented by organization decides when/what to scale in order to increase performance of the cloud. In various situations, monitoring services helps to track load characteristics like average vCPU usage. As soon as threshold is reached for a workload, auto-scaling service is triggered which follows adding of resources and setting load balance preference already pre-planned [54]. On the other hand, if load is dropped to a specific threshold, un-wanted resources and reversing the process is triggered by auto-scaling service. Flawlessly, performance is maintained for a workload of user when auto-scaling service is properly implemented [55]. Usually, the performance of the micro-services can be improved through scaling the resources and their numbers. The former one is known as vertical scaling while the latter one is called horizontal scaling. However, the cloud dynamic workloads consist of bursts which create difficulties in effectively scaling the application. For cloud workloads, it is difficult to identify bursts online to maintain the application performance. In [56], a burst-aware auto-scaling method is proposed to detect burst in cloud workloads through forecasting, resource prediction, and scaling decision-making.

Auto-scaling methods can be triggered either reactively, proactively or in a hybrid way. Few of them are rule-based policies, some use machine learning, analytical queuing, and reinforcement learning methods to scale application and resources. The authors, in [56], have presented a discussion of various machine learning based prediction methods, such as recurrent neural network (RNN) and autoregressive integrated moving average (ARIMA), to improve the micro-services performance. In large, existing methods rely on offline workload data; and this might be a tedious activity. Note that, auto-scaling is largely studied in the context of VMs and cloud workloads; however, scaling micro-services is relatively unexplored. In [57], the authors present an approach to provision resources for micro-services using linear regression along with multi-class classification. Their approach estimates the future workload and CPU demand through CPU usage of the cluster and provision of the resources. Similarly, [58] presents a cost-effective auto-scaling technique for the cloud-hosted micro-services which uses artificial and RNNs to estimate the workload. Further, their technique uses optimization in order to affectively allocate resources for the micro-service applications. Moreover, [59] presents an auto-scaling framework for containerized applications through monitoring utilization levels and statistics of the host resources e.g. CPU. These scaling methods are based on the workload arrival patterns.

Fig. 6 shows a taxonomy of various auto-scaling methods which are classified based on their characteristics. There are other methods that scale the application resources in case the number of connected users exceed certain threshold. This threshold can be either static or dynamic. Dynamic threshold varies time to time and can be estimated various statistical methods [60]. Various approaches use different methodologies to scale either number of VMs or increase the amount of resources to a particular VM. Further, these scaling methods or algorithms can be triggered either automatically (on-demand, periodically) of explicitly (occurrence of an event) just like resource optimization techniques.

7.9. Schedulers

Apart from the above performance techniques, scheduling policies (allocation, placement, and consolidation policies) can be used to take benefits from heterogeneous hardware in terms of performance gains. Rich literature is available in the cloud community that propose different types of scheduling methods [55,61–63]. Largely, they have focused on distributed schedulers — where several schedulers work together through exchanging monitoring information. Hybrid clouds that run various technologies, such as containers, VMs, bare-metal, and nested containers; offer support for different approaches to scheduler. For example, either a single, hierarchical and distributed schedulers can be used on top of hypervisors [64,65]. Note that, single schedulers refer to centralized while the other two refer to decentralized architectures, as shown in Fig. 7. These schedulers can be optimally configured according to resource heterogeneity and workload demand [66]. In [36,64], authors have spent considerable efforts to put various types of schedulers into different taxonomies.

Schedulers are of utmost importance because they affect the cost of operating the cluster. For example, a poor scheduler may result in low resource utilization that costs money because energy and performance efficient servers might be left idle. Moreover, distributed schedulers can be monolithic, two level, shared state, fully distributed, and hybrid by architectural point of view.[11] Besides these, schedulers can also be classified as task scheduler, service scheduler and VM/container scheduler. A task scheduler is responsible to assign a task to a VM/container or hardware for execution. A service scheduler places a particular service over the appropriate resources. A VM/container scheduler refer to allocation policies in the context of cloud computing.

[10] https://www.cloudflare.com/learning/serverless/serverless-performance/

[11] https://www.cl.cam.ac.uk/research/srg/netos/camsas/blog/2016-03-09-scheduler-architectures.html

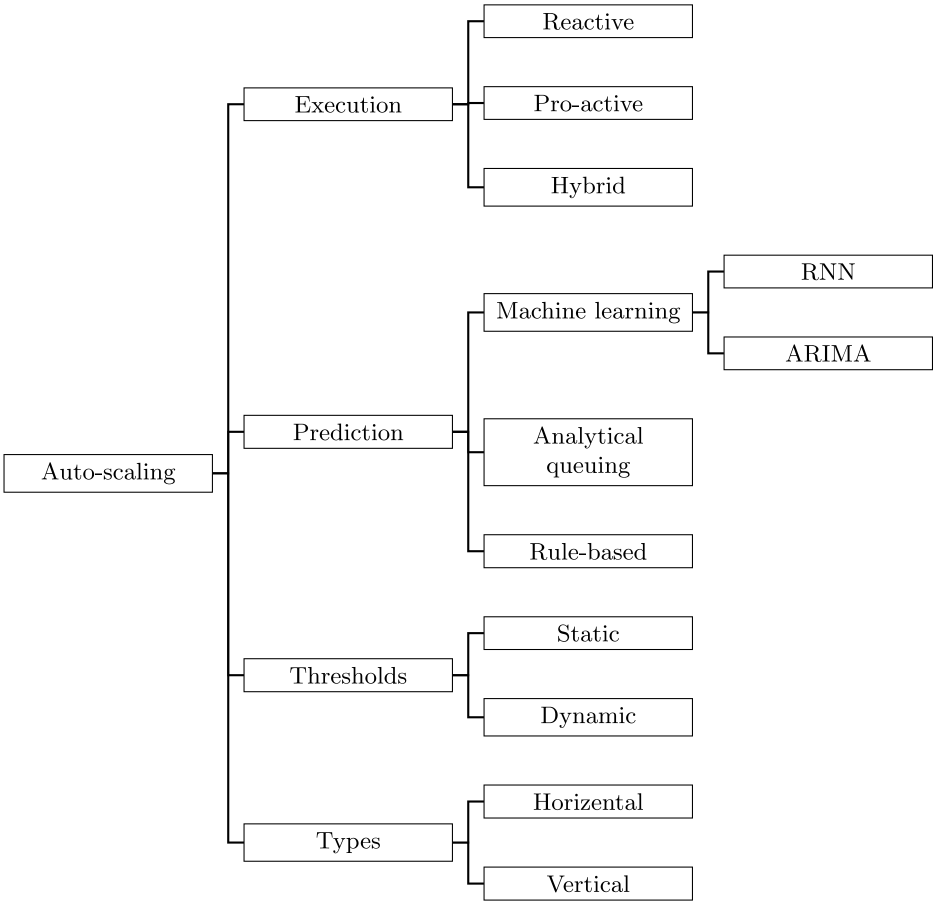

Fig. 6. Taxonomy of auto-scaling approaches.

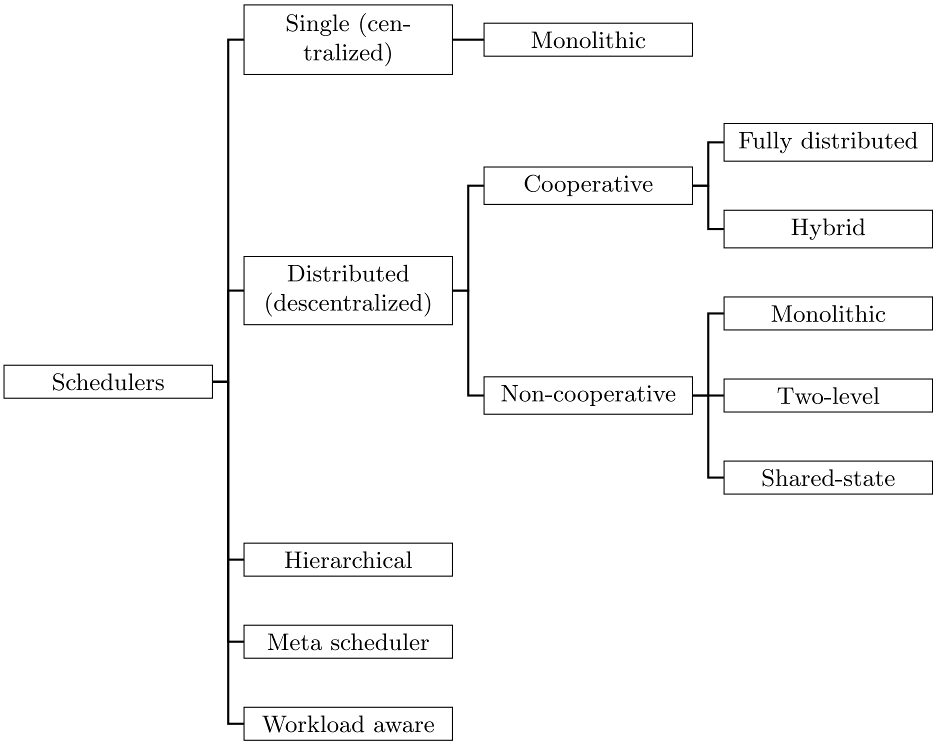

Fig. 7. Taxonomy of various types of schedulers.

7.9.1. Single scheduler

A single scheduler relate to a centralized approach where the entire distributed systems is in its control and have information of all resources. These are also known as monolithic schedulers. This type of scheduling offers robust resource management, possibly due to reconfiguration [66], but it suffers from single point of failure. The well-known Google cluster scheduler "Borg" and Kubernetes scheduler are monolithic [67]. A single scheduler can take more appropriate decisions and, therefore, could reduce the optimization goals (triggering consolidation algorithm to reach certain objectives) which probably increases performance of the applications. Continuous efforts are being made in the existing literature [36,64] to improve the functionality of the schedulers in terms of higher resource utilization levels. From energy consumption point of view, a single resource manager can take more appropriate service placement decisions, as demonstrated in [63]. Significant efforts have been made to utilize various machine learning based resource prediction techniques, which can be integrated into the scheduler to take robust resource placement decisions.

7.9.2. Distributed schedulers

Distributed schedulers offer a decentralized methodology where each scheduler has some knowledge of the cluster resources, but, not essentially [68]. Decentralization can be achieved in two different ways: (a) cooperative; and (b) non-cooperative. Regarding (a), all schedulers has knowledge of the cluster resources, and they share these information with each other to take appropriate decision. This also enables objectives of the various schedulers are met. However, in (b) scheduler do not share cluster information with each other and every scheduler has tendency to optimize its own objective. Since, clouds may be distributed across several geographical areas, and it would not be possible to manage the entire resources using a single manager. This may also refer to various schedulers designed for particular service providers, such as IaaS, PaaS, SaaS, in order to manage multi-provider infrastructure in a multi-access edge computing system. In [36], authors have presented an overview of various scheduling techniques in distributed clouds.

7.9.3. Hierarchical schedulers

In hierarchical system, all schedulers (distributed) are transformed in a tree structure which make it possible that cluster resources, at a particular level, can communicate with other resources in above or below levels. Therefore, through communicating with higher hierarchy resources, high-level scheduling decisions can be made [64]. The term "meta-scheduler" is also used for the high level scheduler that works on top of several schedulers. These schedulers are different from distributed schedulers in the context that all schedulers take placement decisions under the control of a single scheduler.

7.9.4. Meta schedulers

A meta scheduler is the one that takes inputs (placement decisions) from several heuristic-based approaches; and decide appropriate allocation and migration. For example, as described in [69], the max Jobs scheduler is an example of meta scheduler. Largely, heuristic are enough fast to reach an approximate decision rather than a meta-heuristic which may take non-trivial time to reach an optimal decision. There is a vast amount of research that suggest schedulers which can reach optimal decision using methods like particle swarm optimization, simulated annealing, and genetic algorithm, etc. However, an optimal decision can be made only if the workload/tasks are known in advance. The algorithms which are based on this assumption are known as offline algorithms. Unfortunately, cloud workloads is dynamic (online) and it is not reasonable to use such methods. Therefore, other methods like backfilling is used to improve the resource utilization levels of running workloads [70]; and, therefore, energy efficiency.

7.10. Consolidation

Beside workload specific schedulers (brokers), consolidation techniques could also be made enough mature, well-informed of different workloads and their performance over various resources, geographical location, in an IaaS cloud [17]. For example, if a particular workload is performing the worst on a host, then, the application/workload could be migrated to another host, probably a best performing host. However, this will need additional resources e.g. storage server (such as network area storage — NAS) to store collected data and robust machine or deep learning techniques. Imagine hundreds or thousands of cluster nodes which are updating their information on NAS servers, periodically, which will itself generate a lot of traffic and, therefore, burden on the datacentre network. Research shows that consolidation can be expensive in terms of both performance degradation and energy consumption point of view [61,71]. Consolidation involves migrations that could be expensive, particularly, if the migrated workload is terminated by the end user during migration or just after the migration is being completed. Further to energy consumption benefits, providers can achieve performance benefits if they know that a particular VM or container is not performing to its expected level on a particular machine. For example, when similar workloads are co-located on a particular machine; then, while competing for same resources severe performance loss can be observed. In such scenarios, the consolidation technique can be used to re-organize allocation. The optimization can be triggered either periodically or explicitly when certain event occurs. Largely, consolidation is studied in the context of energy savings [22,54,60,72,73], but, its benefits for performance gains are relatively unexplored.

In geographically distributed clouds, certain cluster may be powered using cheaper energy sources, such as renewables, or the electricity prices may be lower than other location. In such circumstances, providers can use consolidation methods to reduce their operational costs. Further, due to the intermittent natures of the renewable, workloads could be migrated across different locations when it is most energy efficient or when enough renewables are available. These decisions are usually taken based on pre-defined threshold levels of machines: (i) if certain machines are under-loaded i.e. their utilization levels are lower than a pre-defined threshold values, then workload from these under-utilized machines can be migrated to somewhere else, and they be put into energy saving modes to reduce energy consumption; and (ii) if certain machines are over-loaded i.e. their utilization levels are exceeding certain pre-defined threshold, then part of workloads can be migrated to somewhere else to reduce performance loss. These thresholds can be either static or dynamic/adaptive — that changes according to resource usage [60]. Dynamic thresholds can result in more energy savings and performance gains. Various consolidation with migration methods are further described in Section 8.2.1.

8. Taxonomy of energy efficient systems

In this section, we describe various approaches for energy deficiency in large scale systems, in particular, clouds and datacentres. Various methods have been identified, and mapped onto taxonomies as shown in Fig. 8. In the rest of this section, we follow this taxonomy for organization and discussion of the paper contents.

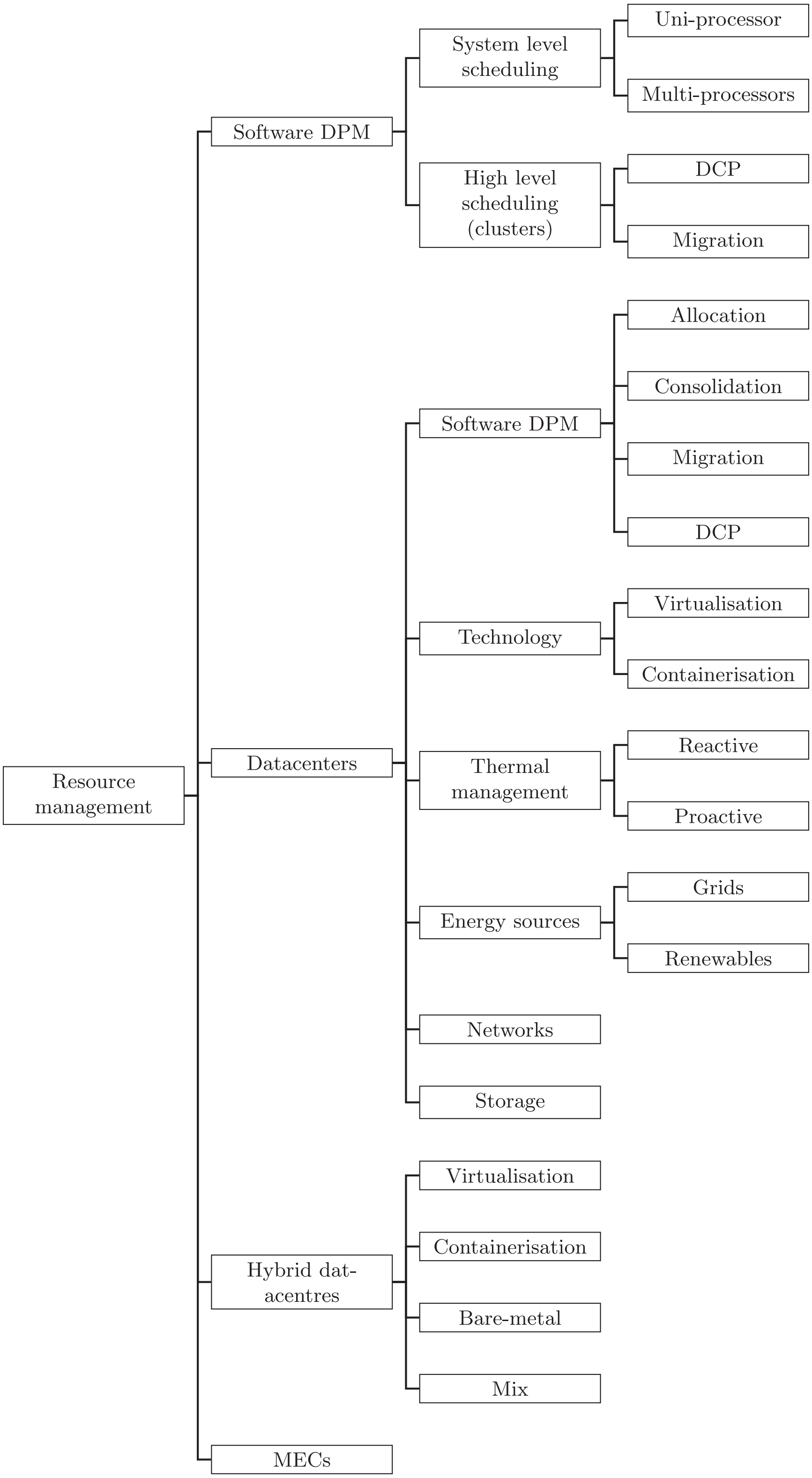

Fig. 8. Taxonomy of power management techniques in computing systems.

8.1. System level scheduling

The scheduling can be defined as the way by which the process is assigned to the specific processor. The system level scheduling is the scheduling of a single system. The scheduling of a system can be performed by time constraint or without time constraint. The time constraint tasks are known as Real Time tasks. The scheduling of these tasks must be performed in such a way that these tasks must be completed with in a particular time frame or deadline. Real time systems are further divided into two types: (i) hard real-time system (*HRTS*); and (ii) soft real-time system (*SRTS*) [35]. The former one deals with strict deadlines whereas the latter one have soft deadlines. Tasks with soft deadlines can be scheduled later even if their deadlines are not met.

The hard real-time systems are those systems in which the tasks are not completed within a time period then some catastrophe will happen. For instance, when there is an auto controlled train, it is not made to stop instantaneously. As the signal goes to red, the train slowly stops over after covering some distance, though it is supposed to stop over at the same point. The reason behind it is the application of the brakes for taking activation to stop. Coverage of the distance is dependent on the speed of the train. It is therefore obvious that speed of the train helps the controller to figure out the time to apply brakes at certain time to stop the train at the right place. The time taken imposes restriction over the response time for the job. It concludes that as soon as red signal is received, then, activation of brakes must be accomplished within certain span of time [74].

The soft real-time system are those systems in which no catastrophe will happen by missing its deadline. The (*HRTS*) are used in time critical systems like missile systems, atomic nuclear plants, air traffic control systems etc. The (*SRTS*) can be used in those systems which are not time critical like online air reservation system, multimedia streams etc. The system may consist of a single processor or multiprocessor. The uniprocessor system exists earlier than the multiprocessor system, so the field of uniprocessor scheduling algorithm is matured as compared to multiprocessor system. The multiprocessor system is complex as compared to uniprocessor system. Due to the simplicity of uniprocessor system there is a lot of optimal uniprocessor scheduling algorithms.

8.1.1. Uniprocessor scheduling

The uniprocessor systems are those systems that consist of a single processor. It is a simple system so there exists a lot of optimal scheduling algorithms [32]. The uniprocessor system consists of a single processor in which the tasks are contained in a single queue [74]. The tasks are contained in a single queue from which it is selected for execution. The selection of tasks from a single queue is very easy. There is a number of uniprocessor scheduling algorithms exist such as first come first serve (FCFS), the shortest process next (SPN), the shortest remaining time (SRT), the shortest job first (SJF), rate monotonic (RM) algorithm, and earliest deadline first (EDF) algorithm [74].

8.1.2. Multiprocessor scheduling

A multiprocessor system consists of more than one processor that execute a particular task in combine [74]. It is a more complex system as compared to a uniprocessor system. The assignment of tasks to more than one processor is a complex job. There are no optimal algorithms for multiprocessor systems. The multiprocessor scheduling algorithms are: *p-fair* scheduling algorithm; energy-aware stochastic task scheduling algorithm; and etc. These algorithms are further divided into: partitioning scheduling algorithms; and global scheduling algorithms. The partitioning scheduling algorithms split all tasks into *n* number of queues and each queue is connected to a specific processor. Then, these tasks are executed from that queue which is connected to a specific processor [74]. Each processor has its own queue, so, the uniprocessor scheduling algorithm is used to execute these tasks. The global scheduling algorithm has, on the other hand, only one queue in which all tasks are contained and stored. The task which has the highest priority in a global queue amongst *n* processors is selected for execution first.

8.1.3. High level scheduling

Resource management in large scale systems such as cluster, grid and cloud is critical. It efficiently uses the resources of large systems and provides the Quality of Service (QoS) to end users. The cluster and grid provides the best effort services to the user. Whereas, the cloud provides reasonable resources to the user [12]. These large scale systems use resource management to solve complex mathematical calculation, strict delay systems and service delivery system. If these systems does not deliver the required performance then the user will not pay for it [75]. The system will deliver the required performance through energy efficient, performance aware resource management and scheduling techniques. The energy and performance aware resource management will allocate the task to that server or Virtual Machine (VM) which is performance and energy efficient. The performance aware resource management will execute the tasks on optimal server which will meet the time and energy requirement of users. The switching on/off system (power cycling) is used for energy saving by switching off the system which is idle [76]. The switching will also consume extra energy by switching on the system again. So the energy saving will be achieved not to switch again and again.

8.1.4. Dynamic Capacity Planning (DCP)

"Capacity planning is the procedure by which infrastructure, application services, and IT resources' procurement are prearranged over an explicit and predefined period of time [12,54]. DCP is an exercise for IT management to foresee and estimate the forthcoming necessities of an enterprise IT platform and its connected crucial entities, components, and services at runtime",[12] The DCP arrange the resources on time to users and not so early that it is unused for long time. It will provide the resources according to resource demand of users. The capacity planning and resource management will make an efficient use of resources by predicting the current and future requirement of users. The large systems just as cluster, grid and cloud shutdown the host instead of switching the system to sleep mode. The main reason of shut down is that when the host is idle it will consume about 60% of the fully utilized host energy [71]. These datacentres are about 20% to 40% utilized which means that a lot of energy is wasted [1]. The energy consumption of the datacentre is minimized by switching off the idle servers. The switching off is also difficult in continuous running server just like the web server. When the host is switched from on to off, this requires time and energy that will translate to an increase in running time and energy consumption [76]. The host is switch on or off in such a way that the energy consumption and the performance of the system is not increased.

The DCP will switch on/off the system according to system situation. When the system is idle then it is switch off. By switching off the energy consumption of the system is reduced significantly, but switching on the system again will also take time and energy. It is very difficult to predict that when the resources of the system is not used or use again in the future [77]. The prediction of resource usage is not possible. So it is impossible to predict to restart the resource on time or to switch off the resource. so to solve this problem the system is operated in sleep mod so that the system is restarted quickly to save energy and time.

8.2. Service placement

In the cloud literature, various approaches to resource/service placement have been suggested, and evaluated, for energy, performance and costs optimization. Largely, the existing methods

[12] https://www.techopedia.com/definition/13932/capacity-planning

assume the placement problem as a kind of bin packing issue. Felter et al. [78] studied resource management of containers (Docker) and VMs (KVM), and compared the achievable performance level of various applications (CPU, memory, storage and network intensive) w.r.t bare-metal. The authors have considered workload metrics such as latency and throughput to determine virtualization and containerization overheads. The interesting point in their experiments is that the execution times for VMs and containers overlap for certain kinds of workload. Moreover, their findings show that "containers and VMs impose almost no overhead on CPU and memory usage; they only impact I/O and OS interaction". Based on their research finding, the authors reject the idea that "IaaS should be implemented on VMs and PaaS on containers" – as there is no technical reason [78]. Unfortunately, migrations are not taken into account.

Scheepers et al. [79] suggest resource isolation as one of the major issues in container-based virtualization. The authors have investigated the performance of hypervisors (Xen) and containers (LXC) for various application types. For a script written in PHP and which inserts randomly generated data into a database, the authors concluded that the same script executes in 16 s on a Xen platform, while it took 335 s on a LXC platform. This shows that LXC are unable to isolate resources successfully as compared to Xen.

Mondesire et al. [80] have investigated the performance of bare-metal, VMs, containers and virtualized containers for executing interactive game-based simulations, individually. The authors suggest that containers performance is comparable to the bare-metal; and mixing containers with VMs offer performance benefits between VMs and containers, alone. Scheduling applications over a VM or on a container that runs inside a VM both have trade-offs that can justify the increased level of resource usage, isolation, ease in deployment, and resource management. However, application migration in terms of energy consumption and performance is not investigated. Mavridis et al. [81] comprehensibly studied resource utilization, containers' isolation, performance and energy consumption, while executing various workloads, when run over bare-metal and inside VMs. Through empirical evaluation, the authors suggest that containers running on KVM consume approximately 7.11% less energy compared to running on Xen hypervisor. Moreover, the workload performance and energy consumption of bare-metal resources is strongly dependent on the workload type. Unfortunately, migrations are not considered.

Lebre et al. [82] have investigated three various VM placement algorithms i.e. Entropy, Snooze and Dvms that fall under categorizes centralized, hierarchical, and distributed scheduling, respectively. Their empirical evaluation demonstrates efficiency of the distributed approach in terms of solving SLA violations quickly. Moreover, a generic tool "VMPlaceS" is suggested in order to evaluate various VM placement and consolidation policies in terms of energy consumption. It is noteworthy to underline that VMPlaceS supports workload fluctuations and datacentre churn (e.g. node removals). However, their results are limited only to the virtualization technology (VMs); and containers, bare-metal or hybrid environments (containers run over bare-metal and/or VMs) are not taken into account.

8.2.1. Consolidation

The virtualization and containerization technologies offer support for consolidating existing workload on fewer resources; thereby, saving energy through DCP. There are a lot of consolidation algorithms that could save energy in one or other way. For example, pMapper [83] comprises a number of resource consolidation methods for heterogeneous virtualized resources. The system accounts for energy cost, migration costs and performance levels while deciding consolidation of VMs and application onto fewer number of hosts. Similarly, Sandpiper [84] is a resource consolidation approach that periodically monitors and detects hotspots, possibly, due to termination of VMs. Furthermore, it tries to reallocate and reconfigure VMs on demand. In order to decide appropriate VMs for migration, the Sandpiper technique sorts all VMs using a single metric based on CPU, network, and memory loads a.k.a volume-to-size ratio (VSR). Then, it prefers to migrate the most loaded VM from an overloaded host to one which has enough unused capacity. Bobroff et al. [85] have proposed the dynamic server consolidation technique that deals with decreasing the SLAs violation and the amount of capacity required by users. The proposed method estimates future resource demand based on historical data and trusts on periodic executions to decrease the number of hosts to provision VMs.

ReCon [86] targets dynamic server consolidation in multi-cluster datacentres. The proposed method deliberates static and dynamic costs of hosts, VM migration costs, and historical resource usage data to offer an optimal dynamic strategy to map VMs to hosts over time. Khanna et al. [87] proposed the migration algorithm. It is triggered when the host is executing more or less number of VMs than its capacity, based on some predefined threshold values. The proposed migration algorithm guarantees to meet all the SLAs. The SLAs in this situation is defined as response time and throughput. The authors have tried to decrease the total number of active hosts and migration cost (performance) to increase the remaining energy of the entire system. These methods account for IaaS energy efficiency but not for performance gains in workloads. Consolidation of tasks in multiprocessor systems can be achieved through task migration; whereas, in virtualized and containerized systems this is supported by VM and container migrations respectively. Note that, the concept of migration is almost the same across these systems. Next, we describe both approaches to migrations in detail.

Pongsakor et al. [88] presented a container re-balancing technique in order to increase the container schedule rate and cluster utilization. Container migration is discussed in the context of a large real dataset from Google. Moreover, a comparison of VMs and containers is presented. However, their approach is based on the assumptions that containers run on the bare metal. Also, the model used to capture the container migration time is rebuttal. Nider et al. [89] have investigated the migration of containerized applications between servers inside a datacentre (heterogeneous), in order to improve power efficiency. A post-copy container migration technique is implemented based on the CRIU technology. Their results demonstrate that the post-copy migration approach significantly reduces container's down-time, and potentially reduces network traffic as well.

Understanding whether migrating a VM, a container or a function running inside a container is very important for energy and performance efficient resource management in a heterogeneous cloud. Consolidation of VMs have been widely explored in the literature [12,71,90], nevertheless, containers on top of bare metal hardware, containers executing within VMs and their consolidation is not enough examined yet. Sareh et al. [91] have investigated containers and their resource management in a containerized cluster. Numerous resource allocation and consolidation policies have been experimentally evaluated in [92] using an event driven cloud simulator — containerCloudSim [93]. In their research, albeit datacentre energy efficiency is explored, however, performance of various workloads (in terms of runtimes) is not discussed. Additionally, the work proposed in [91,92], consolidate either containers or VMs (individually); but, unlike our approach, cannot migrate both, concurrently. Moreover, the resource manager is unable to decide itself whether a container, a VM or both should be migrated. Note that in our current work we have individual brokers to migrate container, VMs or both.

Yong et al. [94] have studied the impact of code migration (offload) on application performance and energy consumption. The presented migration approach minimizes the code offload latency through copying the least and only necessary data. This can be achieved through: (i) identifying only the necessary contexts for method execution; (ii) parsing application binaries in advance; and (iii) applying parsing result to migrate heap data selectively. Ma et al. [95] have used containers live migration to migrate/hand-off a mobile user's service to the nearest edge in a mobile edge cloud platform. A layers-based container migration strategy is proposed that eliminate unnecessary transfers of a redundant portion of the application file system. The proposed technique reduces total transfer size by: (a) moving the base memory image before the hand-off; and (b) moving only the incremental memory difference when container migration starts. Evaluation of their proposed system shows that the service hand-off time is reduced by 56%–80% compared to the state-of-the-art VM hand-off in the context of an edge cloud platform.

In mobile edge clouds (MECs), the key issue is to ensure that customers always receive expected performance levels when they move across different locations — due to proximity. This can be achieved through migrating services between various MECs. Machen et al. [96] have presented a layered framework for migrating active applications which are encapsulated either in VMs or containers. The authors have divided the container state into various layers such as: (i) base layer — that have OS and kernel without any application; (ii) instance layer — that store applications and their running states. The base layer is supposed to be available on every MEC and only the instance layer in migrated to target host. The instance layer is further sub-divided into application layer that have an idle copy of the application itself, and the running state is stored only in the instance layer — allowing live migration i.e. application is copied while the service runs. Their proposed layering approach allows a significant improvement in service performance — reduced downtime. Furthermore, the difference between VM migration and container migration is quantitatively explored. The experimental results show that containers – LXC have a clear advantage over VMs – KVM; in terms of migration time, service downtime (hence performance), and total amount of data transferred during migration. The key reason is that containers are more compact than VMs and the contents in memory of container is largely relevant to the hosted application; whereas the contents in memory of a VM are related to several other processes (background) that may be irrelevant to the migrated application [96–98].

Chenying et al. [99] also suggested a container live migration method known as "logging and replay approach" – which is comparable to the well-known pre-copy VM live migration approach. In the primary stage, a new container is initiated at the destination host and the activity of the source container is monitored, saved on a log file at the source host. Iteratively, the log file is rerun on the destination host, till the file is small enough. Lastly, the container is initiated on the destination host, while its replica on the source host is discarded. The proposed method significantly decreases the container migration time and application downtime. Nadgowda [100] have deliberated live migration of container in more details; and suggested Voyager – a system that combines memory migration (CRIU-based) and data federation competences of union mounts in order to diminish the migration downtime. "With a union view of data between the source and target hosts, Voyager containers can resume operation instantly on the target host, while performing disk state transfer lazily in the background" [100]. All these methods, with the notable exception of [78], have focused on live migration of containers and/or VMs both, therefore resource consolidation; however applications/code migration, heterogeneity of datacentre resources and applications are not addressed. Moreover, resource management in a hybrid cloud platform, which consists of containers and VMs both, is not addressed.

8.2.2. Task migration

Very similar to VM or container migration [54], the migration of tasks in single or distributed system can be used for fault tolerance, minimizing energy consumption and increasing performance of system. In distributed system, when one of the host goes down then the VM are migrated to the other host for fault tolerance. The energy consumption and performance of the system is increased by migrating the VM to energy and performance efficient host. In multiprocessor and multicore systems, the task migration can be used for load balancing [32]. When equal load is distributed across several multiprocessors or cores, then the tasks will be executed in less time and, subsequently, energy consumption is also minimized. The heterogeneous multiprocessor system consist of processors with different vendors, processing speed, power consumption and storage capacities etc. The tasks are migrated from the lower speed processor to the higher speed so that the tasks are executed quickly. There are some tasks in the system which have time constrained on it. These are real-time tasks, with soft deadlines, which will be migrated from slower to higher speed processor. The tasks will be executed faster and the deadline or time constraints are fulfilled.

8.2.3. VM migration

In [63,101], VM and container migrations are covered in detail. However, to understand the contents of next section, it is essential to describe them a bit here. For recent technologies, VM migration is an essential feature provided [54,102]. This facilitates host services to be moved from one host to another host without any service interruption. In beginning for VM migration, stop and copy method was used (offline). The method was employed by stopping the original VM; copy all its content to the destination and afterwards start the new VM. The only advantage of this method was its simplicity, but, performance due to downtime was directly proportional to the amount of memory which was used by the migrated VM. In live migration technique (online), process of shifting a VM from one host to another is carried without disconnection of client application. All memory, storage and network status for VM on host is transferred from source host to destination host. Live migration may happen in three different ways [103].

(I) Pre-copy: In live migration, the very used algorithm is pre-copy live migration technique [104]. It focuses on reducing the downtime for migration through maximizing the memory synchronization from source to the destination host. There are two steps in carrying out this: (a) in the iteration phase, it synchronizes memory between source and destination by iteratively copying memory data from source to destination; and (b) in stop and copy phase, it stops the source VM and copies the remaining pages is carried to VM at destination. Afterwards, the destination host starts the new VM. The process is shown in Fig. 9.

(II) Post-copy: In live migration, the post copy approach initially suspends VM to be migrated at source host, and copy the minimum CPU states along with important libraries to the destination host. Then, it resumes the VM on the target host and memory pages are fetched, on demand, through network from source host to the target host [105]. The downtime is reduced as compared to pre-copy method [104]. Choudhary et al. [106] have demonstrated a systematic critique on various migration algorithms. The process is shown in Fig. 10.

(III) Hybrid: The basic methodology of the hybrid approach is that if only most accessed memory pages (probably those which are frequently accessed or dirtied recently) are moved before the VM resumes its execution at the target host, then VM performance can be improved — as fewer pages need to be fetched on demand from the source host. Actually, hybrid migration can be seen as

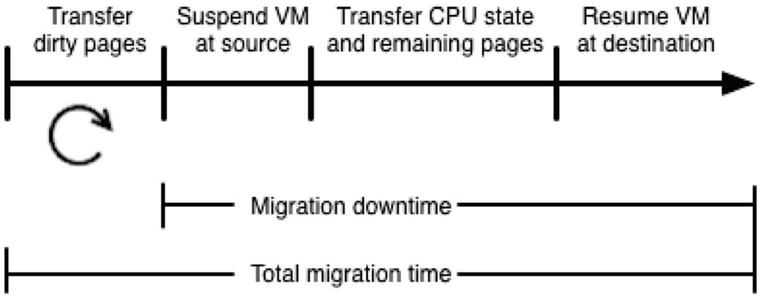

Fig. 9. Pre-copy approach to migration [103].

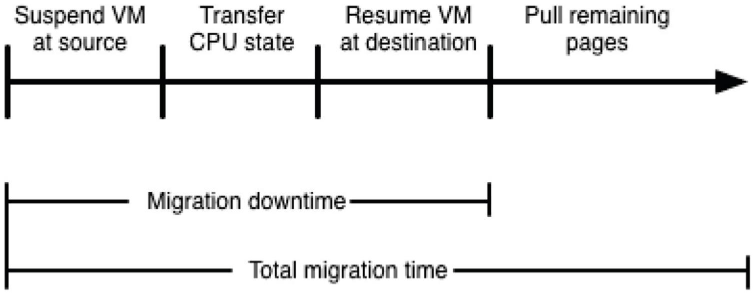

Fig. 10. Post-copy approach to migrations [105].

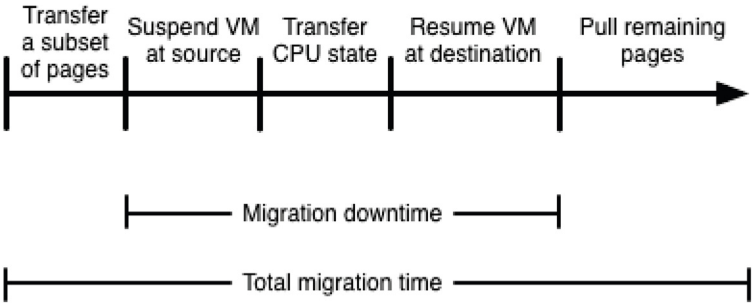

Fig. 11. Hybrid approach to migration [103].

a flavour of the post-copy technique, but, preceded by a limited pre-copy stage [103,105]. The process of a hybrid migration scheme is shown in Fig. 11.

8.2.4. Container migration

Like VM migration, container migration may also occur for several reasons within a datacentre, such as host maintenance, user mobility, electricity price and load balancing. Migrations can still be beneficial where renewable energy is used to decrease datacentre energy costs and CO_2 footprint [21]. Furthermore, if a certain workload performs the worst on a specific host due to co-location or resource heterogeneity, then migrating it to another host could be performance and, hence, cost-efficient. During container live migration, the running container is moved from one host to another. This means migrating data on disk, depending on the underlying method to storage, and memory pages. This leads to two kinds of migration: (i) shared file system, where a container image is run from shared storage such as NFS, GlusterFS, and only memory is copied; and (ii) over-Ethernet migrations, where a container image is run from a local drive (disk), and both memory and disk are copied. Since the container image may itself be large, this latter form of migration may take rather longer. A third kind of migration, particularly for container, is record and replay [107], as shown in Fig. 12.

Unlike VM migration, container migration requires process migration techniques [108], and there may be large amount of OS state associated with the process (e.g. file table, process control block) that must be copied along with the memory pages. On average, container migration takes 10–15 s as demonstrated in footnote.[13] This makes container migrations hard to implement in practice. Certain tools, such as Checkpoint Restart In User-space (CRIU),[14] CMT (container migration tool),[15] and P.Haul,[16] are widely used in industries to checkpoint and restore the container on a target host.

Container migrations are not offered by well-known management frameworks as they are not enough mature yet [109]. In order to consolidate containers without being migrated, an alternative approach is to kill and restart containers. This can be achieved in two different ways reliant on the service type running inside the container: (i) for a stateless container, that runs a stateless service such as RESTful web services, the running service will be affected when the container is killed, but no work needs to be redone; and (ii) for a state-full container, that runs a

[13] https://blog.jelastic.com/2015/12/14/live-containers-migration-across-data-centers-aws-and-azure-integration/

[14] https://criu.org/Main Page

[15] https://github.com/marcosnils/cmt

[16] https://criu.org/P.Haul

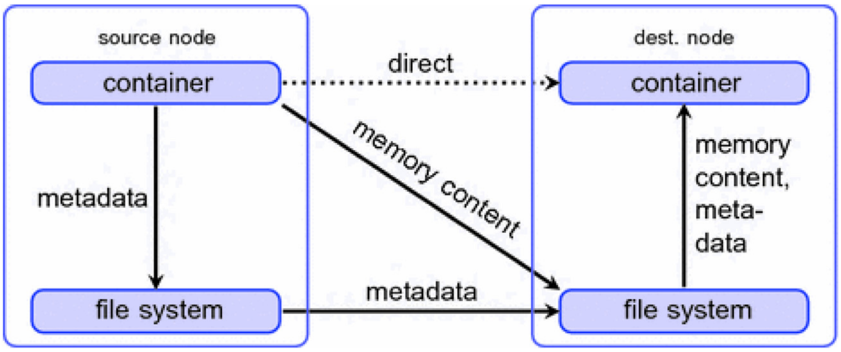

Fig. 12. The process of container migration (using CRIU).[11]

state-full service such as long computation, the aborted segment will need to be re-executed at the target host. Moreover, migration of containers are reliant on several extra libraries and OS kernel characteristics that might not be essentially accessible on the destination host. Therefore, these dependencies could reduce the total number of potential destination hosts to accommodate containers which have been selected for migration. The process of migrating VMs consists of copying application state plus the state of the guest OS (VM) which may lead to lengthier migration periods as compared to container migrations — which only requires copying the application state. Note that, Docker[17] version 1.13 supports container live migration using CRIU[11] technology as demonstrated in footnote.[18] CRIU and P.Haul both checkpoint the container state, migrate/copy the meta-data and restore the meta-data at the destination host using a pre-launched container (similar to the one check-pointed on the source). This can be achieved in two different ways: (i) only memory is migrated — for stateless containers; and (ii) both, memory and the file system (or ephemeral storage) are migrated to the destination — for state-full containers. Therefore, copying only meta-data would take shorter time (migration duration) as compared to copying the meta-data plus the VM|container image (ephemeral storage).

Machen et al. [96] have extended the check-pointing-based migration to a layered one as described in [18]. With the layering technique, the authors are able to live migrate a container — the application layer is copied while the service/container is running. Once migration of the application layer is completed, the application is stopped at source; then data, application status and memory contents in the instance layer are copied to the target server [97]. The proposed layered approach is more efficient than the check-pointing technique in terms of performance gains and short downtime (the duration for which the VM or the container service is not available or not responding to its user).

8.3. Placement with migration

Several studies [110,111] discuss the performance efficiency of containers, VMs, and bare-metal, individually; however, hybrid resource management with respect to energy efficiency is relatively ignored, and this is rarely addressed with the notable exception of [112]. In [112], the authors suggest that bare-metal hardware might offer the highest levels of performance at the lowest energy cost. Nevertheless, the evaluated outcomes take single application (job) into account; and, thus, there outcomes are not ensured as optimal or near to optimal, particularly, if a dynamic system is considered with thousands of VMs, containers, or mix of both.

Moreover, the authors in [108,111] have discussed VMs, containers and virtualized containers i.e. when containers run inside VMs, however, bare-metal and energy efficiency are not investigated. Tay et al. [111] suggest exploring different technologies including VMs and containers, in the context of workload consolidation and migration policies. Sharma et al. [108] evaluated that containers running inside VMs offer performance benefits; and neighbouring containers inside a VM could be trusted, as well. Felter et al. [78] explored resource management of VMs (KVM) and containers (Docker), and compared the obtainable performance of numerous applications regarding bare-metal hardware. Their evaluation shows that for certain types of workloads container's and VM's performance overlaps. Moreover, the authors reject the finding that "IaaS should be implemented on VMs and PaaS on containers" — since there is no technical cause. Unfortunately, service migrations are not taken into account. Mondesire et al. [80] have investigated the performance of bare-metal, VMs, containers and virtualized containers for executing interactive game-based simulations, individually. The authors suggest that containers performance is comparable to the bare-metal; and mixing containers with VMs offer performance benefits between VMs and containers, alone. However, scheduling in a hybrid platform and consolidation of workloads in terms of energy consumption are not investigated.

Tchana et al. [113,114] observed that VMs are, largely, underutilized in certain cloud datacentres; and proposed a solution called software consolidation. Software consolidation accommodate several applications, dynamically, on the same VM to minimize the number of used VMs. Moreover, software consolidation can be combined with VM consolidation that reduces the number of hosts in use. Software consolidation could reduce: (i) energy consumption of a private cloud; and (ii) the number of used VMs, therefore, costs, in a public cloud. Their investigation suggests that approximately 40% energy could be saved through software consolidation in their private cloud. Furthermore, approximately 40.5% user's monetary cost could be saved in the Amazon EC2 public cloud. This work closely resembles our consolidation technique; however, their algorithm is very straightforward that could not decide effective migration among VMs, containers and software/applications in terms of energy consumption and workload performance.

8.4. Energy efficient resource management

In this section, we provide a review of various power management techniques which are used to achieve energy efficiency in large-scale cloud datacentres. These include software-based DPM techniques that use different scheduling and consolidation policies at IaaS level. Consolidation is a method in which efficient use of IT related infrastructure is achieved through collecting servers or simply replacing the legacy servers with virtual system

[17] https://www.docker.com/

[18] https://www.youtube.com/watch?v=izycGffZOtg

(server virtualization, containerization) in the cloud premises. It primarily consolidates hardware for more efficient use. Not only consolidation is performed, but it also boosts efficient use of CPU power and memory (storage).

8.4.1. Software DPM

Software-based DPM approaches consist of different resource allocation, placement and consolidation with migration algorithms. These methods can be used to achieve energy and performance efficiency in two different ways: (a) activate hardware capabilities, such as DVFS [32], ALR [33], if feasible; and/or (b) decide appropriate resources to place, migrate, and run workloads. Works including [9,18,101], have discussed various approaches to energy efficiency which are based on software-based DPM methods. The work presented in [22] discusses methods to enhance energy efficiency for computation and network resources being part of large-scale distributed systems using dynamic voltage and frequency scaling (DVFS), or by consolidation (cluster VMs to the fewest hosts in order to elude power). In terms of computational resource, the method works at several levels i-e, from single node to whole infrastructure through which an advantage may be taken in the form of virtualization. However, the reduced energy usage yields limited affect as it decreases performance. This also leads to have a low expected gain for reduction in energy efficiency; as users are directed to use further compute resource if are cheaper in cost. Therefore, it is not a proper solution for rapidly growing large-scale datacentres where compute power is considered keeping reduced carbon emissions.

8.4.2. Technology

Technological growth in datacentres has significantly improved resources' management from energy and performance efficiencies point of view. For example, containerization have almost replaced virtualization technology due to its rapid deployment. From management point of view, difficulty is increased along with better opportunities for appropriate placement and consolidation with migration [63]. For example, various workloads will have different requirements; some of them will perform best in containers, others in VMs and bare-metal [81]. This will soon transform the datacentre industry into hybrid-structure with performance benefits. Apart from these, development in energy sources, hardware design, microprocessor technology, and cooling mechanisms have tremendously improved energy datacentre energy efficiency.

8.4.3. Thermal management

Due to growing energy costs and huge energy consumption of present cooling infrastructures in datacentres, smart and energy efficient cooling systems or management techniques (i.e. thermal management) have been proposed in the literature. Datacenter cooling infrastructure consumes nearly the same amount of energy as the computing infrastructure. The Arrhenius' equation applied to microelectronics states that every 10°c increase in device temperature doubles its failure rate that could significantly increase datacentre's TCO. Thermal management is divided into two different approaches: (i) reactive; and (ii) proactive [115]. The former one keeps datacentre temperature at a predefined threshold; and the later one dynamically adjusts the temperature using load management policies. Switching on/off resources may create hotspot that could increase cooling costs along with performance loss [116]. Renewables may decrease cooling costs, particularly, in geographically distributed datacentres — through migrating the workloads. Orgerie et al. [22] have discussed various energy efficient cooling techniques such as free cooling, temperature-aware scheduling and thermal management. Free cooling uses outside air or sea water to cool infrastructure that leads providers to install their resources in regions with cold climate.

Table 4
Energy efficient cooling techniques at datacentre level.

Cooling Technique	Approach	Implementation
Sensor nodes	monitor resource temperature and adjusts air-conditioning	LBNL, HP (DSC)
HACS and CACS	reduce hot and cold air mixing [containment system]	HACS could save 43% annual cooling costs; 15% reduced PUE compared with CACS
Smart water use	sea water	Google
Liquid cooling	replace processor fans with liquid	Asetek
Free cooling	use outside air	AST Modular, IBM
Fan control	reduce face speed just like DVFS	system level technique

Another approach is fan control that monitors processor temperature and, through thermal conditions, adjusts the intensity of active coolers to save power [117]. The goal can be achieved by adjusting the fan speed according to the processor heating. Though this technology was initially aimed to decrease noise of active heat sink, but it is highly related to system power consumption. For instance, if a fan is running on 5 volts, it will consume 83% less power in contrast to running on 12 volts. Dynamic Smart Cooling (DSC) uses wireless sensors to continuously monitor resource temperature and adjust air-conditioning accordingly, in HP clusters [116]. This approach could save up to 30%–60% cooling energy costs. Moreover, Google reports a 40% reduction in their cooling costs through the use of machine learning techniques on sensors collected resource usage data.[19] Furthermore, rack management and their positions could also affect datacentre energy efficiency. Note that inadequate or faulty cooling can result in resource overheating which affects datacentre reliability, workload performance and device lifetime. ASHRAE[20] suggests that raising the datacentre temperature up to one degree could also save significant energy costs. Various energy efficient cooling techniques in cloud datacentres are summarized in Table 4.

8.4.4. Energy sources

Renewable energy sources, such as solar, wind, play a major role in revolutionizing the datacentre energy efficiency. Renewable sources help in reducing the environmental impacts. A report on Google datacentres shows that in 2017 and 2018, Google was able to run their infrastructure 100% on renewables,[21] as shown in Fig. 13. Moreover, Amazon AWS[22] and Microsoft[23] have almost exceeded 50% goals of achieving renewables usage to operate their infrastructure in 2018. However, renewable sources are intermittent and may not be available at certain time. Therefore, grid energy cannot be completely replaced. This is evidenced by the fact that in 2018, approximately 63.5% of electricity generation in the United States was from fossil fuels such as coal [118]. As discussed before, being efficient at all cost may

[19] https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/

[20] https://blog.schneider-electric.com/datacenter/2013/02/01/4-ways-to-reduce-energy-consumption-in-any-data-center/

[21] https://www.blog.google/outreach-initiatives/sustainability/100-percent-renewable-energy-second-year-row/

[22] https://aws.amazon.com/about-aws/sustainability/

[23] https://venturebeat.com/2019/04/15/microsoft-says-it-now-uses-60-renewable-energy-to-power-its-data-centers/

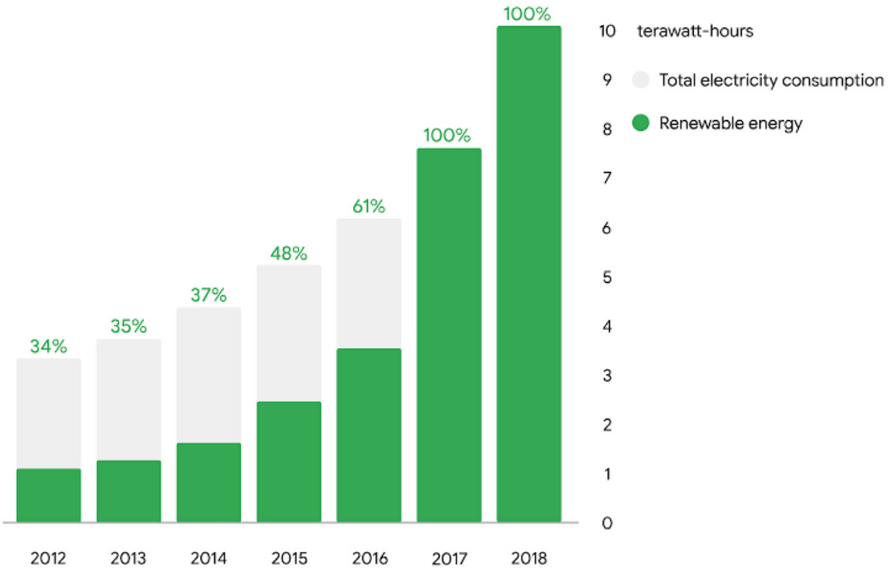

Fig. 13. Google's total energy consumption versus renewable purchase.[8]

be even counter-productive both economically and ecologically; particularly when the power supply is varying due to renewable energy sources. Furthermore, apparently energy efficient hardware may mean that certain workloads need to run for longer, and the trade-off between efficiency and runtime, as well as the implication in cost of increased runtime, all need to be addressed.

Operating datacentres at different energy sources would need proper workload allocation techniques so that workloads could be placed on appropriate infrastructure. Subsequently, consolidation policies would help in moving workloads across infrastructure power by various energy sources, when a particular source is not available and vice versa. For example, when renewable is not available, then workloads on renewable's powered resources would be instantly moved to resources powered by grid energy. Furthermore, service providers can increase their profits and economics through placing and migrating workloads onto renewables, where feasible, rather than on grid sources. A framework proposed by Li et al. [119], discusses green datacentres power management. The work primarily targets design of green datacentre having power provided by various sources. The decision on power management is taken on the availability of power output for base-load, UPS storage capacity, performance of the job and variability in renewable power. An increase in runtime for job by 3%, maintaining anticipated energy usage efficiency, and a 12% increase in UPS backup time is achieved.

The problem with contracting energy is that it is sort of cheating. Whilst renewable energy is being generated somewhere, that may not be where your datacentre is. The third option is to fix this — deploy renewable sources of energy on the local grid providing power 24/7, so the datacentre will actually consume renewables at all times. This is much more difficult because of the varying location of datacentres and weather (intermittent). Some facilities are located in regions with abundant wind, solar and/or hydro while others are not. Google began work to achieve 24/7 renewables in 2018. Furthermore, their approach towards carbon-intelligent computing[24] offers ways to shift workloads to times of day with peak renewable energy. It is drawing closer to the development of its claim, or contacting to third parties, sources for renewable energy that go specifically into the local network. Google published an article about their approach which incorporates a few interesting illustrations of the concept[2]. However, this is still not possible to switch all datacentre operations to renewable; because in 2018, approximately 63.5% of electricity generation in the United States was from fossil fuels such as coal [118]. Furthermore, various regions offer different prices for energy consumption. These will make the resource providers to run user workloads competitively for cheap energy sources and low prices to in crease their money savings and ecological impacts. However, this should be optimized subject to network costs in terms of latencies and workload performance i.e. execution times (translating to user bills). This needs further exploration, investigation and research which is the focus of this paper [120].

In [121], a new scheduling method for energy sources is formulated to enhance usage of renewable energy, and then considers reducing energy obtained from conventional grid and battery backup. The dynamic method encompasses to use grid power covering energy. The main advantage of this method is that it is evidently realistic to ponder supply of energy to a datacentre from the grid, though it has limitation to implement dynamic power. On contrary, it is also tried to optimize usage of battery by boosting low capacity of the batteries. The given algorithm gives high efficiency in case of renewable energy being efficiently and exhaustedly exploited by using workload scheduling. Furthermore, Liu et al. [122] integrates workload management for datacentres by taking gains of efficiency made available by changing demand which exploits variations in time for electricity's price, renewable energy availability (due to intermittent nature), and efficient cooling. There are two phases in the design i-e, first important feature is integrating three main silos of datacentres: IT, power and cooling. Secondly, a mix of theory, modelling, and implementation. The core of the design is focused on optimizing cost solved through workload management. The method depicts reduction of grid electricity use by 60% having no impact over the quality of service provided by the applications.

A VM placement method by Khosravi et al. [72] discusses energy reduction and carbon costs for datacentres placed geographically but with the limitation that all the locations are residing within the same country. A carbon footprint management approach by Doyle et al. [123] discusses only load balancing but consideration over renewable energy is not focused. A Parasol and GreenSwitch scheme proposed by Goiri et al. [124] takes a prototype system where dynamic scheduling is enabled for

[24] https://www.blog.google/outreach-initiatives/sustainability/100-percent-renewable-energy-second-year-row/

workloads and different energy sources are selected. Not like the work presented in [125], this work also considers servers at the same location. In comparison to existing work presented, the approach in [125], gives workload shifting in order to schedule workloads across various datacentres. The main objective of their work is to minimize overall carbon footprint as well as making sure that the average response time of the requests is intact. Along with these, their objective is also focused on geographically placed datacentres at different time zones having various carbon concentrations as well as renewable energy availability. Deng et al. [126] introduced an on-line control strategy which helps to reduce operational power cost as well as energy usage from non-renewable sources by 60%. This is achieved by applying a two-step Lyapunov optimization method and, further, a control pseudo code is designed. The decisions are made by complete utilization of renewable energy, power acquisition on real-time basis and live charging/discharging of uninterrupted power supply (UPS) without keeping intact the dynamics of the system. The balance between operational cost, availability of datacentre and UPS lifetime is achieved. The said method is robust on stability of the system in terms of time varying demand and supply for power.

With 640 datacentre outages in the UK alone in 2015 and outages expected to be more common in near future,[25] there is a need at least for proper capacity planning, consolidation of workloads onto servers powered by renewables, and migration of workloads when it is most energy, and therefore cost, efficient, to safeguard supply and reduce the drain on renewable generation and storage equipment. Further to previous discussion, the location where electricity/energy is produced is really very important in terms of its ecological and environmental efficiency. For instance, to produce 1.0 KWh of energy, it generates approximately 3.0 grams of CO_2 in Norway, 100.0 grams in France, 600 grams in Virginia, and 800 grams in New Mexico. Therefore, switching amongst various energy sources offers environmental sustainability as well.

8.4.5. Networks

The use of computer networks consume a high quantity of electricity due to the increase of devices like laptop, cellphone, personal computer and servers. According to a report, worldwide Internet traffic has almost tripled between 2015 and 2019; and is further estimated to double by 2022.[26] These large number of devices are connected to the Internet which will consume a lot of energy [127]. For example in the US, network consumes about 1% to 3% of the total electricity consumption [13]. The networks along with other devices connected to it consume as much as the energy consumed by the datacentres. Various studies show that by applying energy saving techniques in networks, the energy consumption of datacentres can be reduced significantly. The tremendous use of artificial intelligence techniques, interest of folks in streaming videos, migration to private clouds, and huge number of sensors to be deployed in the emergent era of IoT (internet of things) create questions on expected growth in datacentres' energy consumption.

With the rising electrification and connectedness of the society including smart phones, laptops, ad-hoc and Wi-Fi networks, a report [14] has suggested that 1% to 3% of US electricity use comes from compute network devices (voice, data) and large system fixed networks. This figure is even larger if energy use also accounts for the consumer devices (cellphones, tablets etc.) and wireless networks. In [128], it is estimated that access networks including consumer devices use as much energy as datacentres, and these have a faster growth rate of usage. Table 5 shows the estimated energy usage of network equipment in Italy by 2020 — where due to scaling effect, minimizing the energy consumption of less energy hungry devices (home networks) can still lead to higher savings per year than other devices [129]. The studies in [25,128] also signifies the need of energy efficient techniques to diminish the energy consumed by access networks in clusters and grids. In [130], it is estimated that a typical compute cluster network accounts for 30% of the total energy consumed — including 15% for access switches, 10% for aggregate switches and 5% for core switches. The typical cluster [130] consists of total 1536 nodes, 512 access, 16 aggregation and 8 core switches, and is 30% utilized (average).

Table 5
Networks energy consumption forecast 2015–2020 [one billion kWh is 1000 GWh and one GWh is one million kWh] [129].

Network type	Energy consumption (W)	No of devices	Energy consumption (billion kWh/year)
Home	10	17,500,000	1.533
Access	1280	27,344	0.307
Transport	6000	1750	0.092
Core	10,000	175	0.015
Total			1.947

Table 6
Networks energy consumption (in TWh).

Year	2015	2018	2021	
			Moderate	High
Fixed	74	89	111	78
Mobile	118	172	169	118

Table 6 summarizes the energy consumption of fixed and mobile networks with future estimates [11]. Both fixed and mobile networks consumed approximately 261 TWh of energy which is roughly equal to 1.1% of the worldwide energy demand. If we, now, assume moderate level of energy efficiency techniques (10% improvements) in networks, then a significant consumption is estimated in 2021. However, high level of energy efficiency would possibly break the estimated growth tend. Mobile networks are rapidly shifting towards more efficient 4G, 5G (and eventually 6G networks); and it is expected that by 2022, emergent 4G, 5G and 6G eras will be responsible to hold 83% of the total mobile traffic, as compared to 2G shares with less than 1% [11,13].

8.4.6. Storage

In a single server, storage disk is often considered as second high energy consuming device [131]. Furthermore, [131] also suggests datacentre storage consumes approximately 2% of the overall energy consumption. Large scale systems such as clusters, grids and clouds consist of several modules, as shown in Table 7; and storage is one of them which consumes a lot of energy. Due to increasing demand for streaming videos, energy consumption of storage devices is expected to grow in near future. In datacentres, storage is either locally present in the form of hard disks, tapes, and SSDs; or it has a network of several commodity storage devices known as Storage Area Network (SAN) or Network Area Storage (NAS) [12]. In 2012, most datacentres reported 10%–24% annual growth in data storage. Furthermore, a report on storage energy efficiency suggests that 1 Watt hour of energy savings in storage might result in approximately 1.9 Watt hour energy savings at datacentre level.[27]

[25] http://www.greendatacenternews.org/articles/share/887707/

[26] https://www.comsoc.org/publications/tcn/2019-nov/energy-efficiency-data-centers

[27] https://www.energystar.gov/products/data_center_equipment/data_center_storage

Table 7
Energy consumption of different devices in a typical datacentre [131].

Device	Energy consumption (%)
CPUs	61.0
DRAM	18.0
Networking	5.0
Storage	2.0
Power overhead	7.0
Cooling	3.0
Miscellaneous	4.0

Various techniques, such as data de-duplication, data compression, thin provisioning and storage virtualization, will help to improve storage systems; and possibly reduce purchase of storage equipments. Besides these, datacentres have their own policies for insertion and retrieval of data from storage equipments. The report indicates that, in clusters, storage module can consume about 27% of the total energy use. In recent years, the rapid increase in smart devices and online services have drastically increased the amount of storage; that subsequently increase in data will increase the energy consumption of large scale systems. In the literature, several mechanisms are used to decrease the energy consumption of storage systems while ensuring high availability of stored data. For example, energy consumption of the disk drive and storage could be minimized through SPM and DPM level techniques. Furthermore, high capacity disk drives have also significantly reduced the energy consumption of storage as compared to ordinary disks. The Classic Serial Advanced Technology Attachment (SATA) drives can save energy up to 50% as compared to capacity fibre channel drives per Terabyte (TB). Sold State Drives (SSDs) could provide better performance than traditional disk drives, but they are too expensive to buy. Using recommended storage devices from Energy Star[28] in datacentres could save significant amount of energy and, thus, money.

8.5. Hybrid clouds (virtualization, containerization, bare-metal)

In major IT companies such as Google, Rackspace and Amazon AWS, virtualization and containerization technologies are usually used to execute customers' workloads and applications. Since long, Google run users' applications in containers, Rackspace offer bare-metal hardware, whereas AWS run them either in VMs (EC2), containers (ECS) and/or containers inside VMs (Lambda); therefore, making resource management a tedious activity. The role of a resource management system is of the greatest importance, principally, if IT companies practice various kinds of sand-boxing technologies, such as bare-metal, VM and/or containers, in their datacentres (hybrid platforms) to offer quality services to customers. The absence of a single, workload-aware resource manager creates questions on datacentres energy efficiency and performance of the workloads. There is a need to pact with this challenge through advising a reference architecture and a single, platform-independent, resource manager. The challenge should be to determine the right abstractions to enable the design of an integrated service leveraging the core mechanism — without the implementation of dedicated services, such as individual scheduler and platform-specific monitoring, for each kind of sand-boxing technology [63].

Moreover, similar workloads may perform quite differently on various platforms; some of them may execute faster on containers than VMs, some will perform worse on containers than bare-metal and vice versa [78,132]. Similarly, certain cloud users may need full access to bare-metal resources in order to get total control on their provisioned hardware. This is also evidenced through the recent introduction of AWS bare-metal instances; that will, probably, soon force IaaS (Infrastructure as a Service) providers to rethink of using various platforms in their datacentres. Therefore, complexities would arise when the cloud provider uses a mix of these sand-boxing technologies — in order to maximize their resource usage and reduce their operational costs. In such scenario, certain workloads may perform the worst on VMs, but, would execute faster over the containers, virtualized containers or bare-metal (non-virtualized) platforms [133]. Lower execution times may mean higher energy efficiency, lesser users costs and vice versa; however, energy efficiency may also relate to the sand-boxing technologies, workload types and hardware energy consumption profiles.

This creates opportunities for hybrid clouds that implement various sand-boxing technologies, such as the Intel's CIAO (Cloud Integrated Advances Orchestrator),[29] Magnum,[30] kolla[31] and, later on, appropriate workload placement and migration decisions are made. The goal could be achieved through clustering the available resources such that each cluster corresponds to a particular sand-boxing technology. Furthermore, each cluster may have either its own scheduler or share a centralized scheduler. Using individual schedulers for each sand-boxing technology such as containerization, virtualization, virtualized containers, bare-metal may not be appropriate in terms of energy efficiency and performance; due to the lack of entire datacentre state and resource usage information at each scheduling (platform) level. If these schedulers, that belong to various platforms and sand-boxing technologies, can communicate and share entire datacentre state with each other (i.e. centralized scheduler); appropriate energy and performance efficient workload placement and migration decisions could be taken [63]. In [81], the authors have studied the combination of virtualization and containerization technologies through running containers on top of VMs. Their aim is to improve containers' key problem i.e. isolation (since containers share the same kernel) and, to incorporate benefits of containers into VMs. Therefore, containers were run on bare-metal, and inside VMs (using hypervisors KVM, and Xen); and the authors suggest its possibility subject to trivial performance overhead. Besides high resource utilization, their evaluation suggests that running containers on KVM is more energy-performance efficient than running them on Xen. Unfortunately, their work has ignored resource management aspects such as scheduling, consolidating workloads, in hybrid clouds. Moreover, migrations are not examined.

Besides VM placement [101], container placement is also largely studied in the literature. For example, in [134] authors have presented ECSched, a graph-based scheduler to handle concurrent container requests in heterogeneous clusters subject to multi-resource constraints. The ECSched scheduler assumes a batch of requests, at the same time, to find a condensed placement. The authors suggest that ECSched produces good results, in terms of low completion times, and improved resource utilization. However, the proposed scheduler is impractical for online problems when tasks do not arrive in batches; or, the online problem should be converted to an offline problem in order to fetch requests at the same time. Moreover, VMs, nested containers, hybrid platforms and their energy efficiency is not explored. Similarly, KEIDS [135] incorporate a container scheduler/management system on top of Kubernetes to account for interference and energy consumption of IoT applications in distributed clouds (operated by different energy sources). The KEIDS scheduler is

[28] https://www.energy.gov/eere/femp/purchasing-energy-efficient-data-center-storage

[29] https://ciao-project.github.io/

[30] https://wiki.openstack.org/wiki/Magnum

[31] http://docs.openstack.org/developer/kolla/

approximately 14.42%, and 31.83%, better that the FCFS scheduler in terms of improved energy utilization, and minimal interference. In [136], a communication-aware worst fit decreasing heuristic algorithm is proposed for container placement. Moreover, a container reassignment strategy is presented to balance the containers distribution across various servers and optimize application performance and throughput. Albeit, renewable, and distributed clusters are taken into account; however, except containers, other scenarios such as VMs, virtualized containers, hybrid platforms and migrations are not explored.

ProCon [137] schedules containers subject to: (i) the instant resource utilization of hosts; and (ii) estimation of future resource usage. The ProCon scheduler balances the resource contentions across the cluster and reduces task runtimes through monitoring their execution progress. ProCon reduces completion time by up to 53.3% and improves performance by 23.0% against the default scheduler available in Kubernetes. Various approaches to virtualization (full — KVM, para — Xen and OS level — Docker) are discussed in [138]. The performance of KVM and Docker was compared in three different ways: (a) the CPU and memory usage of the host, (b) Idleness of CPU, memory usage and I/O performance through migrating a large file, and (c) performance of the Web server through JMeter. These comparisons show that Docker is faster than KVM. The authors have only compared KVM and Docker which were configured on a single host. Moreover, the authors demonstrate that placement algorithms affect the performance of VMs and containers. The PIVOT task scheduler [139] supports cross-cloud, cross-region execution of data-intensive applications while hiding the complexity of the underlying heterogeneous systems and respecting cost and performance requirements of the containerized application. PIVOT has two capabilities: (i) an application scheduler schedules various tasks of an application; and (ii) the global scheduler has a task queue that put tasks for final dispatching and placement onto hosts. Furthermore, the scheduling problem is modelled as a vector bin-packing problem and solved effectively using greedy approximation algorithms such as first fit heuristic.

In [140], a task-oriented and energy-aware scheduler "HEATS" is suggested for containerized workloads that allows customers to trade performance vs. energy needs and exploits the resource heterogeneity. In the first phase (probing), HEATS learns the energy and performance characteristics of hosts. In the second phase (monitoring), it monitors tasks execution on hosts. In the third phase (scheduling), HEATS speculatively migrates workload across various hosts to match customers' demands. Their evaluation suggests that, depending on the workload type, HEATS can save up to 8.5% energy while marginally affect the overall task runtimes (by at most 7%). Renewables along with appropriate resource allocation and consolidation approaches can mitigate the energy related issues in cloud environment. In [141], containerized workloads are placed on those clusters which has enough renewable energy. Moreover, a container consolidation scheme is designed to minimize the energy consumption of hosts. In [142], authors have discussed bin-packing, approximate and meta-heuristic algorithms. Moreover, a container scheduling approach is suggested to account for various objectives such as load-balancing and multi-resource guarantee. Other works have also suggested meta-heuristic based approaches to solve the workload placement problem [143–145]. However, [142] suggests that meta-heuristic approaches can take hours to reach a solution, and, are not suitable for container scheduling.

8.6. Mobile edge clouds or multi-access edge computing

Real-time applications such as on-line gaming and video conferencing have extremely on-demand requirements to provide high-quality results within the agreed time e.g. shorter response time through closer communication with the application server. Using cloud platform to deploy real-time applications offers several benefits including reduced OpEx (operation costs) and on-demand resource allocation — assign resources per needs of the application. However, real-time applications may become sensitive to the quality of network e.g. latency between users and services. Therefore, the real-time applications' requirements could possibly be addressed through the emerging edge/fog computing technology — allows computations to be accomplished at the edge of the network. The rationale of commissioning this technology is to allow services allocated within the proximity of customers and closer to where computational results are desirable. This can be achieved through deploying small-scale or micro datacentres closer to customers, and connected to original cloud datacentres [44,96].

Management of resources in mobile edge clouds is very challenging, because offering quality services to the end-users depends on various players, with moderately conflicting goals, such as infrastructure owners (IaaS), network operators, and application providers (SaaS), where each player controls a particular part of the system. Integral to the problem is the facts that both communication and computation capacity is needed to guarantee high QoS in terms of low response time and high throughput. Existing works either assume that the whole infrastructure is managed by a single player, largely, the resource providers, or separate the management of the network resources from edge computing capacity. Resource allocation could be assumed as a multi-objectives optimization problem. Since, various service providers might have different objectives, and, mostly they might be competitive towards achieving their goals; therefore, appropriate management techniques are needed. For example, [146] suggests game-theoretical approaches to balance the trade-off amongst various providers and their competitive gaols.

Besides missing their management systems, there are certain questions that needs to be investigated. For example; where these datacentres should be deployed; which services should be installed; where and how resources should be allocated to users' applications; how user mobility should be handled; and how the system should be optimized to minimize various objectives. Major use cases for fog technology include: intelligent agriculture, healthcare, smart cities and traffic management. Some of these will be of utmost importance to service providers in the era of new digitization. The mobile, telecommunication and other IT companies would directly take benefits and more profit through introducing more reliable services to their customers. Subsequently, it will affect business revenue and nation's economics. Furthermore, through efficient management of the available resources, this is possible to decrease the energy and network usage of large infrastructure. These savings in energy can either be translated to more profit, reduce customer costs, reinvest in the business and/or household etc.

Gillam et al. [44] have also discussed VMs, containers and code/functions (Function as a Service — FaaS) in order to explore edge computing for on-vehicle and off-vehicle computation that will be needed to support connected and automated/autonomous driving. To minimize end-to-end latency, the authors suggest that it is essential that computation should be more local to vehicles. However, vehicle mobility will create opportunities for application/code migration, and with the notable exception of [96], it is rarely discussed in the literature.

9. Areas in need of further research

Besides rich literature and state-of-the-art resource management methods within the field of energy, performance and cost aware cloud computing [3,9,61,147], we believe that certain aspects should still be further addressed. This statement applies to both service placement and migration-based consolidation scenarios in various types of heterogeneous datacentres. Below, we describe what gaps we have identified for improvements and how these gaps will be considered for further investigation and research.

9.1. Efficient allocation and placement

Performance of workloads and applications are strongly dependent on the hardware which they run. When heterogeneity is considered, certain workloads may run energy, performance efficiently on certain hardware; however, other workloads may not. Moreover, technologies like virtualization and containerization have different overheads which may offer various levels of energy and, therefore, performance efficiencies. Therefore, workload specific allocation techniques should be developed, investigated and implemented. For example, workloads that may perform best on containers might be run over containers; and whose performance is ensured in VMs should be run in VMs. The other two options: (i) run workloads over bare-metal hardware; and (ii) run containers in VMs; would create opportunities and possible design of hybrid clouds. These hybrid clouds would require a decentralized approach to placement methodology i.e. single and distributed schedulers.

9.1.1. Single scheduler

A single scheduler can be a single point of failure [63]; however, cost-efficient allocation decisions can be triggered if it is made aware of the whole infrastructure (available resource characteristics such as energy consumption, workloads performance details and so on). This needs further research and investigation in the context of a large and adaptable cloud service.

9.1.2. Distributed scheduler

A distributed scheduling approach consists of several schedulers that cooperate with each other to take appropriate placement decisions. Albeit, failure of a single scheduler will affect certain services only; however, management will be a tough activity [64]. Both approaches have their own advantages and disadvantages. There is a need to investigate the impacts of using a single, centralized and a distributed, decentralized (meta) scheduler in terms of energy consumption, workload performance and users costs.

9.2. Consolidation with migration

Containers are smaller (lightweight) than VMs, therefore, their migrations would take shorter durations than VMs. However, due to their small sizes, more containers would fit into a single server (bin-packing problem). Since, the server performance is affected in proportional to the number of VMs and/containers they accommodate. This means that, besides small migration durations, containers performance will be highly affected than VMs. This trade-off between VMs and containers migration needs further exploration. Moreover, containers will often migrate even quicker than VMs; and each migration costs. Therefore, there will be situations in which even migrations will be worst that no migration. Moreover, migrations not only target energy efficiency, hardware maintenance; but, it is possible to migrate workloads if their performance is not ensured on certain hardware. Similarly, in hybrid clouds (multiple platforms such as bare-metal, virtualized, containerized and nested — containers over VMs) there will be multiple options for migrations during the consolidation rounds. Single entities could be migrated in inter-platform consolidation; and multiple entities can be migrated in intra-platform consolidation. The former one will create more gaps (stranded resources) than the later one. Further research is essential to investigate challenges related to workload consolidation [148].

9.2.1. VM

VM migration is well-studied in the existing cloud literature [60,71,149]. However, migrations costs energy and results in performance degradation that can be as high as 10%. Moreover, VM migration in the context of heterogeneous hardware may even be worse than no migration [76]. In the literature, migrations are considered for energy efficiency but not for performance improvement.

9.2.2. Container

Container can be quickly migrated, but, it is possible that their migration efforts are wasted — when a container is terminated during the migration process or just after its migration is completed. If we perform migrations for the purpose of energy efficiency or performance improvements, then, we have to ensure that our efforts are not wasted. Therefore, container migrations are needed to be investigated not only for energy efficiency but for performance gains as well [30,150,151]. For example, if workloads run slow on certain servers; then, it is possible to migrate them to other resources. Note that, containers can be migrated from bare-metal (containerization) and/or from VMs.

9.2.3. Application

Another option is to migrate bare-metal applications or VMs that consist of several containers. From energy consumption point of view, various application migrations techniques (software consolidation) are discussed in [113,114]. From performance perspective, this needs further research.

9.3. Workload aware consolidation

As discussed earlier, consolidation of VMs is largely studied for energy efficiency. However, migrations could be triggered for performance gains or, at least, to ensure SLA. Therefore, if schedulers are made aware of various types of workloads, performance impacts, resource heterogeneities; then, it is possible to take appropriate energy, performance and cost-efficient migration decisions. For example, if a particular application perform worse on a specific server [17]; then that application will be migrated to such a server than ensures its performance needs. From implementation point of view, this may need appropriate hardware/storage servers and various prediction techniques [152,153]. Moreover, how energy efficiency and performance of the workload would be affected using various options. We believe, these kinds of options and their impact on energy, performance and user costs are not investigated in the existing literature.

9.4. Energy source and price aware resource management

Largely, cloud service providers (CSP) use various sources to produce electricity that fuel their infrastructure, offices, cooling, and lighting devices etc. Furthermore, a single CSP may have different infrastructure or datacentres which are distributed over various geographical locations (e.g. the notion of availability zones in the Amazon web service cloud). Different energy sources and, as well as, geographical locations would have different prices for electricity at different times of the day.[32] To face and solve these challenges, the design and implementation of an effective,

[32] https://datacenterfrontier.com/google-shifting-server-workloads-to-use-more-renewable-energy/

and elastic scheduler and resource management approach to monitor the whole infrastructure is difficult, yet also essential. A scheduler is an integral and main part of a resource management system which is responsible to schedule jobs/VMs on appropriate resources [72,154].

9.5. From cloud to the multi-access edge computation

Real-time applications such as online gaming, healthcare, intelligent agriculture, and video conferencing have on-demand requirements to provide quality results within the agreed time [155]. Using cloud platform to deploy similar applications offers several benefits including reduced operational costs, on-demand service allocation and management [156]. However, these applications may become sensitive to the quality of networks. Therefore, their service requirements could, possibly, be addressed through the emerging fog computing technology. The rationale of commissioning this technology is to allow services allocated within the proximity of customers and closer to where computational results are desirable. This could be achieved through deploying small-scale datacentres closer to customers and connected to original cloud datacentres through dedicated or third-party networks.

Besides missing their management systems, there are certain questions that needs to be investigated. For example; where these datacentres should be deployed; which services should be installed; where and how resources should be allocated to users' applications; how user mobility should be handled; and how the system should be optimized to minimize various objectives [157]. Major use cases for fog technology include: intelligent agriculture, healthcare, smart cities and traffic management.

9.6. Mobility management in multi-access edge computing

Connected, or more formally, autonomous cars [44] are considered as mitigators of issues such as traffic congestion, road safety, inefficient fuel consumption and pollutant emissions that current road transportation system suffers from. Multi-access edge computing systems (MECs) offer cloud computing capabilities closer to the radio access network (RAN) in 4G and 5G telecommunications and converge with other radio access technologies such as WiFi or Satellite. An MEC can be seen as a cloud server running at the edge of a mobile network which is deployed and executed over functions, containers, and virtual machines (VMs). A cloudlet is like an MEC which implies several servers, providing compute services to connected users in their close proximities. Therefore, when users move across several MECs, it would be essential to migrate their applications transparently.

9.7. Competitive service placement among various providers

The Internet of Things (IoT) is producing an extraordinary volume of data daily, and it is possible that the data may become useless while on its way to the cloud for analysis, due to longer distances and delays. Fog/edge computing is a new model for analysing and acting on time-sensitive data (real-time applications) at the network edge, adjacent to where it is produced. The model sends only selected data to the cloud for analysis and long-term storage. Furthermore, cloud services provided by large companies such as Google, can also be localized to minimize the response time and increase service agility. This could be accomplished through deploying small-scale datacentres (referred to by name as cloudlets) where essential, closer to customers (IoT devices) and connected to a centralized cloud through networks — which form a multi-access edge cloud (MEC) [146,158].

The MEC setup involves three different parties, i.e. service providers (IaaS), application providers (SaaS), network providers (NaaS); which might have different goals, therefore, making resource management a difficult job. In the literature, various resource management techniques have been suggested in the context of what kind of services should they host and how the available resources should be allocated to customers' applications, particularly, if mobility is involved. However, the existing literature considers the resource management problem with respect to a single party. It is essential to look for appropriate resource management techniques with respect to all three parties i.e. IaaS, SaaS, NaaS; while meeting their objectives, simultaneously [159,160].

9.8. Renewables and carbon intelligent computation

The world is on track for perilous climate alter, having about misplaced room to assist contamination within the mix of gasses that make up the air. In spite of a rise in clean, renewable energy supplies in certain nations like the UK, Germany; and a fractional move from coal to natural gases in other countries, the worldwide GHG contamination still proceeds to rise — and at an expanding pace within the most later a long time [61]. This alarms the need for energy-aware computation to be taken into account on a priority basis without any negative impact on applications' performance [54]. The renewable energy has reached up to approximately 54 TWh (3.3%) of the Britain's total energy utilization in 2010, having expanded consistently since 2005; and by approximately 15% from 2008 to 2009. We will expect, through these figures, more than a four times increment in the renewable energy utilization by 2020; in the event that approximately 15% of the energy requirements are to be met from renewable energy sources. The utilization of renewable energy will ought to rise by ∼17% annually to meet these objectives. A large proportion of datacentre usage, a main source of energy consumption, today is through the use of public clouds. Furthermore, it is estimated that in 2021, approximately 53% of worlds' all servers will be located in the hyper-scale public cloud datacentres. This basically means Amazon Web Services (AWS), Google compute cloud, Facebook, and Microsoft Azure [1].

The problem with contracting energy is that it is sort of cheating. Whilst, renewable energy is probably being generated somewhere, that may not be where your datacentre is located. A potential option to fix this issue is to deploy renewable sources of energy on the local grid providing power 24/7 a week; so that the datacentre can actually consume renewables at all times. This is much more difficult because of the varying locations of datacentres and unpredictable weather conditions (intermittent). Albeit, some IaaS facilities are located in regions with abundant renewables such as wind, solar and/or hydro while others are not. The Google team began work to achieve 24/7 a week available renewables in 2018. Furthermore, their approach towards carbon-intelligent computing[33] offers ways to shift workloads to times of day with peak renewable energy. It is drawing closer to the development of its claim, or contacting to third parties, sources for renewable energy that go specifically into the local network. Google published an article about their approach which incorporates a few interesting illustrations of the concept[33]. However, this is still not possible to switch all datacentre operations to renewable; because in 2018, approximately 63.5% of electricity generation in the United States was from fossil fuels such as coal [118]. Furthermore, various regions offer different and varying prices for energy consumption. These will make the resource

[33] https://www.blog.google/outreach-initiatives/sustainability/100-percent-renewable-energy-second-year-row/

providers to run user workloads competitively for cheap energy sources and low prices to increase their money savings and ecological impacts. However, this should be optimized subject to network costs in terms of latencies and workload performance i.e. execution times (translating to user bills). This needs further exploration, investigation and research which is the focus of this paper [120].

10. Conclusion

In this paper, we debated on the energy consumption and performance problems in cloud datacentre; and their economical, as well as, ecological impacts. Moreover, we deliberated the energy efficiency of large-scale, heterogeneous IaaS clouds at three different levels: (a) hardware design; (b) resource management techniques like allocation, consolidation, migration; and (c) applications development. Besides energy consumption of datacentres, performance of applications and hardware resources was studied in detail. This paper provided a comprehensive explanation of energy, performance, and cost-efficient resource management methods in large-scale, heterogeneous IaaS clouds, and numerous taxonomies were presented to organize and categorize them for further investigation and research. Following are few of the most important messages from the arguments made in this paper:

- internet traffic, number of mobile subscribers and demand for streaming videos are continuously increasing;
- energy consumption of datacentres along with networks, storage devices are expected to grow further in the near future;
- fossil fuels have negative impacts on our environment and global warming;
- due to future workloads' requirements, the cloud service model will shift to a hybrid-type architecture;
- improvements in renewables will increase IaaS energy efficiency, but, complexities would rise in resource management;
- existing consolidation techniques largely account for energy efficiency, but, not for performance gains; and
- existing migration methods account for migration costs in terms of energy, but, performance with respect to users costs (runtime), and recovery of all costs, is relatively unexplored.

The contents of this paper will benefit our readers in finding further gaps amongst what previously exists in current systems (IaaS heterogeneous clouds) and what is needed, with the intention that outstanding research questions can be recognized. We trust that more and significant efforts are still desirable to further examine and investigate certain verdicts and outcomes which were offered in this paper. For example, emerging systems such as cloudlets, edge/fog computing and mobile edge clouds (MECs) also trigger the need for migration techniques, particularly, to run user's application at the edge of the network. One major issue that comes with proximity is how to ensure that customers always receive good or expected level of performance as they move across different locations. In the future, we intend to extend this work for such scenarios [96].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported, in part, by the Abdul Wali Khan University, Mardan (AWKUM), Pakistan and, in part, by the Higher Education Commission (HEC), Pakistan. This work is part of a PhD research conducted at the Department of Computer Science, AWKUM, Pakistan.

References

[1] A. Shehabi, S.J. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, E. Masanet, D. Sartor, M. Herrlin, W. Lintner, United States Data Center Energy Usage Report, Lawrence Berkeley National Laboratory, Berkeley, California, 2016, p. 4, LBNL-1005775.
[2] Muhammad Zakarya, Energy, performance and cost efficient datacenters: A survey, Renew. Sustain. Energy Rev. 94 (2018) 363–385.
[3] Muhammad Zakarya, Lee Gillam, Energy efficient computing, clusters, grids and clouds: A taxonomy and survey, Sustain. Comput.: Inform. Syst. 14 (2017) 13–33.
[4] Chen Wei, Zhi-Hua Hu, You-Gan Wang, Exact algorithms for energy-efficient virtual machine placement in data centers, Future Gener. Comput. Syst. 106 (2020) 77–91.
[5] Irfan Mohiuddin, Ahmad Almogren, Workload aware vm consolidation method in edge/cloud computing for iot applications, J. Parallel Distrib. Comput. 123 (2019) 204–214.
[6] Najet Hamdi, Walid Chainbi, A survey on energy aware vm consolidation strategies, Sustain. Comput.: Inform. Syst. 23 (2019) 80–87.
[7] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Gener. Comput. Syst. 25 (6) (2009) 599–616.
[8] Rajkumar Buyya, Satish Narayana Srirama, Giuliano Casale, Rodrigo Calheiros, Yogesh Simmhan, Blesson Varghese, Erol Gelenbe, Bahman Javadi, Luis Miguel Vaquero, Marco A.S. Netto Netto, et al., A manifesto for future generation cloud computing: research directions for the next decade, ACM Comput. Surv. (CSUR) 51 (5) (2018) 105.
[9] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, Albert Zomaya, et al., A taxonomy and survey of energy-efficient data centers and cloud computing systems, Adv. Comput. 82 (2) (2011) 47–111.
[10] Mohammad Aldossary, Karim Djemame, Ibrahim Alzamil, Alexandros Kostopoulos, Antonis Dimakis, Eleni Agiatzidou, Energy-aware cost prediction and pricing of virtual machines in cloud computing environments, Future Gener. Comput. Syst. 93 (2019) 442–459.
[11] Arman Shehabi, Sarah J. Smith, Eric Masanet, Jonathan Koomey, Data center growth in the united states: decoupling the demand for services from electricity use, Environ. Res. Lett. 13 (12) (2018) 124030.
[12] Muhammad Zakarya, Lee Gillam, Energy and Performance Aware Resource Management in Heterogeneous Cloud Datacenters (Ph.D. thesis), University of Surrey, 2017.
[13] S.J. Cisco, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, Vol. 2017–2022, Cisco Public Information, San Jose, CA, USA, 2017.
[14] Jonathan Koomey, Growth in data center electricity use 2005 to 2010, in: A Report by Analytical Press, Completed at the Request of the New York Times, Vol. 9, 2011.
[15] Christopher Stewart, Kai Shen, Some joules are more precious than others: Managing renewable energy in the datacenter, in: Proceedings of the Workshop on Power Aware Computing and Systems, IEEE, 2009, pp. 15–19.
[16] John O'Loughlin, Lee Gillam, Performance evaluation for cost-efficient public infrastructure cloud use, in: International Conference on Grid Economics and Business Models, Springer, 2014, pp. 133–145.
[17] John O'Loughlin, A Workload-Specific Performance Brokerage for Infrastructure Clouds (Ph.D. thesis), University of Surrey, 2018.
[18] Ayaz Ali Khan, Muhammad Zakarya, Rahim Khan, Izaz Ur Rahman, Mukhtaj Khan, et al., An energy, performance efficient resource consolidation scheme for heterogeneous cloud datacenters, J. Netw. Comput. Appl. 150 (2020) 102497.
[19] Thang Le Duc, Rafael García Leiva, Paolo Casari, Per-Olov Östberg, Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey, ACM Comput. Surv. (CSUR) 52 (5) (2019) 1–39.
[20] Muhammad Zakarya, Lee Gillam, Energy efficient computing, clusters, grids and clouds: a taxonomy and survey, Sustain. Comput.: Inform. Syst. 14 (2017) 13–33.
[21] Junaid Shuja, Abdullah Gani, Shahaboddin Shamshirband, Raja Wasim Ahmad, Kashif Bilal, Sustainable cloud data centers: a survey of enabling techniques and technologies, Renew. Sustain. Energy Rev. 62 (2016) 195–214.

[22] Anne-Cecile Orgerie, Marcos Dias de Assuncao, Laurent Lefevre, A survey on techniques for improving the energy efficiency of large-scale distributed systems, ACM Comput. Surv. 46 (4) (2014) 47.

[23] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities, in: High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on, IEEE, 2008, pp. 5–13.

[24] Hameed Hussain, Saif Ur Rehman Malik Malik, Abdul Hameed, Samee Ullah Khan, Gage Bickler, Nasro Min-Allah, Muhammad Bilal Qureshi, Limin Zhang, Wang Yongji, Nasir Ghani, et al., A survey on resource allocation in high performance distributed computing systems, Parallel Comput. 39 (11) (2013) 709–736.

[25] Junaid Shuja, Sajjad A. Madani, Kashif Bilal, Khizar Hayat, Samee U. Khan, Shahzad Sarwar, Energy-efficient data centers, Computing 94 (12) (2012) 973–994.

[26] Tarandeep Kaur, Inderveer Chana, Energy efficiency techniques in cloud computing: A survey and taxonomy, ACM Comput. Surv. 48 (2) (2015) 22.

[27] Fahimeh Alizadeh Moghaddam, Patricia Lago, Paola Grosso, Energy-efficient networking solutions in cloud-based environments: A systematic literature review, ACM Comput. Surv. 47 (4) (2015) 64.

[28] Mahmut Kandemir, Shekhar Srikantaiah, Compiler-driven energy efficiency, in: The Green Computing Book: Tackling Energy Efficiency at Large Scale, 2014, p. 43.

[29] Wissam Chedid, Chansu Yu, Survey on Power Management Techniques for Energy Efficient Computer Systems, Laboratory Report, Mobile Computing Research Lab, 2002.

[30] Zeineb Rejiba, Xavier Masip-Bruin, Eva Marín-Tordera, A survey on mobility-induced service migration in the fog, edge, and related computing paradigms, ACM Comput. Surv. 52 (5) (2019) 1–33.

[31] Cheol-Ho Hong, Blesson Varghese, Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms, ACM Comput. Surv. 52 (5) (2019) 1–37.

[32] Nasro Min-Allah, Hameed Hussain, Samee Ullah Khan, Albert Y. Zomaya, Power efficient rate monotonic scheduling for multi-core systems, J. Parallel Distrib. Comput. 72 (1) (2012) 48–57.

[33] Kashif Bilal, Samee U. Khan, Sajjad A. Madani, Khizar Hayat, Majid I. Khan, Nasro Min-Allah, Joanna Kolodziej, Lizhe Wang, Sherali Zeadally, Dan Chen, A survey on Green communications using Adaptive Link Rate, Cluster Comput. 16 (3) (2013) 575–589.

[34] Yuqing Qiu, Chung-Horng Lung, Samuel Ajila, Pradeep Srivastava, Experimental evaluation of lxc container migration for cloudlets using multipath tcp, Comput. Netw. 164 (2019) 106900.

[35] Nasro Min-Allah, Samee U. Khan, Xiuli Wang, Albert Y. Zomaya, Lowest priority first based feasibility analysis of real-time systems, J. Parallel Distrib. Comput. 73 (8) (2013) 1066–1075.

[36] Mohit Kumar, Subhash Chander Sharma, Anubhav Goel, Santar Pal Singh, A comprehensive survey for scheduling techniques in cloud computing, J. Netw. Comput. Appl. 143 (2019) 1–33.

[37] Stephen P. Crago, John Paul Walters, Heterogeneous cloud computing: The way forward, Computer 48 (1) (2015) 59–61.

[38] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, Michael A. Kozuch, Heterogeneity and dynamicity of clouds at scale: Google trace analysis, in: Proceedings of the Third ACM Symposium on Cloud Computing, ACM, 2012, p. 7.

[39] Fei Xu, Fangming Liu, Hai Jin, Athanasios V. Vasilakos, Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions, Proc. IEEE 102 (1) (2014) 11–31.

[40] John O'Loughlin, Lee Gillam, Re-appraising instance seeking in public clouds, in: Science and Information Conference, Vol. 2015, SAI, IEEE, 2015, pp. 807–815.

[41] Fei Xu, Fangming Liu, Hai Jin, Heterogeneity and interference-aware virtual machine provisioning for predictable performance in the cloud, IEEE Trans. Comput. 65 (8) (2016) 2470–2483.

[42] John O'Loughlin, Lee Gillam, Sibling virtual machine co-location confirmation and avoidance tactics for public infrastructure clouds, J. Supercomput. 72 (3) (2016) 961–984.

[43] H. Zhuang, X. Liu, Z. Ou, Aberer. K., Impact of instance seeking strategies on resource allocation in cloud data centers, in: 2013 IEEE Sixth International Conference on Cloud Computing, 2013, pp. 27–34.

[44] Lee Gillam, Konstantinos Katsaros, Mehrdad Dianati, Alexandres Mouzakitis, Exploring edges for connected and autonomous driving, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2018, pp. 148–153.

[45] Susan J. Fowler, Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization, O'Reilly Media, Inc., 2016.

[46] Li Wu, Johan Tordsson, Erik Elmroth, Odej Kao, Microrca: Root cause localization of performance issues in microservices, in: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2020, pp. 1–9.

[47] István Pelle, János Czentye, János Dóka, Balázs Sonkoly, Towards latency sensitive cloud native applications: A performance study on aws, in: 2019 IEEE 12th International Conference on Cloud Computing, CLOUD, IEEE, 2019, pp. 272–280.

[48] Hyuck Han, Young Choon Lee, Woong Shin, Hyungsoo Jung, Heon Y. Yeom, Albert Y. Zomaya, Cashing in on the cache in the cloud, IEEE Trans. Parallel Distrib. Syst. 23 (8) (2011) 1387–1399.

[49] Asaf Cidon, Assaf Eisenman, Mohammad Alizadeh, Sachin Katti, Dynacache: Dynamic cloud caching, in: 7th {USENIX} Workshop on Hot Topics in Cloud Computing, HotCloud 15, 2015.

[50] Eric Jonas, Johann Schleier-Smith, Vikram Sreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar, Joao Carreira, Karl Krauth, Neeraja Yadwadkar, et al., Cloud programming simplified: A berkeley view on serverless computing, 2019, arXiv preprint arXiv:1902.03383.

[51] Lee Gillam, Will cloud gain an edge, or, closer, to the edge, in: International Conference on Cloud Computing and Services Science, Springer, 2018, pp. 24–39.

[52] Garrett McGrath, Paul R. Brenner, Serverless computing: Design, implementation, and performance, in: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops, ICDCSW, IEEE, 2017, pp. 405–410.

[53] Wes Lloyd, Shruti Ramesh, Swetha Chinthalapati, Lan Ly, Shrideep Pallickara, Serverless computing: An investigation of factors influencing microservice performance, in: 2018 IEEE International Conference on Cloud Engineering, IC2E, IEEE, 2018, pp. 159–169.

[54] Tiago C. Ferreto, Marco A.S. Netto, Rodrigo N. Calheiros, César A.F. De Rose, Server consolidation with migration control for virtualized data centers, Future Gener. Comput. Syst. 27 (8) (2011) 1027–1034.

[55] Gunjan Khanna, Kirk Beaty, Gautam Kar, Andrzej Kochut, Application performance management in virtualized server environments, in: Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, IEEE, 2006, pp. 373–381.

[56] Muhammad Abdullah, Waheed Iqbal, Josep Lluis Berral, Jorda Polo, David Carrera, Burst-aware predictive autoscaling for containerized microservices, IEEE Trans. Serv. Comput. (2020).

[57] Hanieh Alipour, Yan Liu, Online machine learning for cloud resource provisioning of microservice backend systems, in: 2017 IEEE International Conference on Big Data, Big Data, IEEE, 2017, pp. 2433–2441.

[58] Issaret Prachitmutita, Wachirawit Aittinonmongkol, Nasoret Pojjanasuksakul, Montri Supattatham, Praisan Padungweang, Auto-scaling microservices on iaas under sla with cost-effective framework, in: 2018 Tenth International Conference on Advanced Computational Intelligence, ICACI, IEEE, 2018, pp. 583–588.

[59] Fan Zhang, Xuxin Tang, Xiu Li, Samee U. Khan, Zhijiang Li, Quantifying cloud elasticity with container-based autoscaling, Future Gener. Comput. Syst. 98 (2019) 672–681.

[60] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Gener. Comput. Syst. 28 (5) (2012) 755–768.

[61] Muhammad Zakarya, Lee Gillam, Managing energy, performance and cost in large scale heterogeneous datacenters using migrations, Future Gener. Comput. Syst. 93 (2019) 529–547.

[62] Anton Beloglazov, Rajkumar Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurr. Comput.: Pract. Exper. 24 (13) (2012) 1397–1420.

[63] Ayaz Ali Khan, Muhammad Zakarya, Rahim Khan, H ^{2}—a hybrid heterogeneity aware resource orchestrator for cloud platforms, IEEE Syst. J. 13 (4) (2019) 3873–3876.

[64] Luiz F. Bittencourt, Alfredo Goldman, Edmundo R.M. Madeira, Nelson L.S. da Fonseca, Rizos Sakellariou, Scheduling in distributed systems: A cloud computing perspective, Comp. Sci. Rev. 30 (2018) 31–54.

[65] Adrien Lebre, Jonathan Pastor, Anthony Simonet, Frédéric Desprez, Revising openstack to operate fog/edge computing infrastructures, in: Cloud Engineering (IC2E), 2017 IEEE International Conference on, IEEE, 2017, pp. 138–148.

[66] Rui Han, Chi Harold Liu, Zan Zong, Lydia Y. Chen, Wending Liu, Siyi Wang, Jianfeng Zhan, Workload-adaptive configuration tuning for hierarchical cloud schedulers, IEEE Trans. Parallel Distrib. Syst. 30 (12) (2019) 2879–2895.

[67] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, John Wilkes, Large-scale cluster management at google with borg, in: Proceedings of the Tenth European Conference on Computer Systems, ACM, 2015, p. 18.

[68] Malte Schwarzkopf, Andy Konwinski, Michael Abd-El-Malek, John Wilkes, Omega: flexible scalable schedulers for large compute clusters, in: Proceedings of the 8th ACM European Conference on Computer Systems, 2013, pp. 351–364.

[69] Ohad Shai, Edi Shmueli, Dror G. Feitelson, Heuristics for resource matching in intel's compute farm, in: Workshop on Job Scheduling Strategies for Parallel Processing, Springer, 2013, pp. 116–135.

[70] Dan Tsafrir, Yoav Etsion, Dror G. Feitelson, Backfilling using system-generated predictions rather than user runtime estimates, IEEE Trans. Parallel Distrib. Syst. 18 (6) (2007) 789–803.

[71] Muhammad Zakarya, Lee Gillam, An energy aware cost recovery approach for virtual machine migration, in: International Conference on the Economics of Grids, Clouds, Systems, and Services, Springer, 2016, pp. 175–190.

[72] Atefeh Khosravi, Lachlan L.H. Andrew, Rajkumar Buyya, Dynamic vm placement method for minimizing energy and carbon cost in geographically distributed cloud data centers, IEEE Trans. Sustain. Comput. 2 (2) (2017) 183–196.

[73] Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, Ammar Rayes, An energy-efficient vm prediction and migration framework for overcommitted clouds, IEEE Trans. Cloud Comput. (2016).

[74] Ayaz Ali Khan, Abid Ali, Muhammad Zakarya, Rahim Khan, Mukhtaj Khan, Izaz Ur Rahman, Mohd Amiruddin Abd Rahman, A migration aware scheduling technique for real-time aperiodic tasks over multiprocessor systems, IEEE Access 7 (2019) 27859–27873.

[75] Hameed Hussain, Saif Ur Rehman Malik, Abdul Hameed, Samee Ullah Khan, Gage Bickler, Nasro Min-Allah, Muhammad Bilal Qureshi, Limin Zhang, Wang Yongji, Nasir Ghani, Joanna Kolodziej, Albert Y. Zomaya, Cheng-Zhong Xu, Pavan Balaji, Abhinav Vishnu, Fredric Pinel, Johnatan E. Pecero, Dzmitry Kliazovich, Pascal Bouvry, Hongxiang Li, Lizhe Wang, Dan Chen, Ammar Rayes, A survey on resource allocation in high performance distributed computing systems, Parallel Comput. 39 (11) (2013) 709–736.

[76] Muhammad Zakarya, An extended energy-aware cost recovery approach for virtual machine migration, IEEE Syst. J. 13 (2) (2018) 1466–1477.

[77] https://www.youtube.com/watch?v=7MwxA4Fj2l4. (Online; Accessed 3 October 2015).

[78] Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio, An updated performance comparison of virtual machines and linux containers, in: Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium on, IEEE, 2015, pp. 171–172.

[79] Mathijs Jeroen Scheepers, Virtualization and containerization of application infrastructure: A comparison, in: 21st Twente Student Conference on IT, Vol. 21, 2014.

[80] Sean C. Mondesire, Anastasia Angelopoulou, Shehan Sirigampola, Brian Goldiez, Combining virtualization and containerization to support interactive games and simulations on the cloud, Simul. Model. Pract. Theory 93 (2019) 233–244.

[81] Ilias Mavridis, Helen Karatza, Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing, Future Gener. Comput. Syst. 94 (2019) 674–696.

[82] Adrien Lebre, Jonathan Pastor, Anthony Simonet, Mario Südholt, Putting the next 500 vm placement algorithms to the acid test: The infrastructure provider viewpoint, IEEE Trans. Parallel Distrib. Syst. 30 (1) (2019) 204–217.

[83] Akshat Verma, Puneet Ahuja, Anindya Neogi, PMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), LNCS, vol. 5346, 2008, pp. 243–264.

[84] Timothy Wood, Prashant Shenoy, Arun Venkataramani, Mazin Yousif, Sandpiper: Black-box and gray-box resource management for virtual machines, Comput. Netw. 53 (17) (2009) 2923–2938.

[85] Norman Bobroff, Andrzej Kochut, Kirk Beaty, Dynamic placement of virtual machines for managing sla violations, in: 2007 10th IFIP/IEEE International Symposium on Integrated Network Management, IEEE, 2007, pp. 119–128.

[86] Sameep Mehta, Anindya Neogi, ReCon: A tool to recommend dynamic server consolidation in multi-cluster data centers, in: NOMS 2008 - IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubiquitous Networks and Services, 2008, pp. 363–370.

[87] G. Khanna, K. Beaty, G. Kar, a. Kochut, Application performance management in virtualized server environments, in: 2006 IEEEIFIP Network Operations and Management Symposium NOMS 2006, Vol. 20, No. D, 2006, pp. 373–381.

[88] U. Pongsakorn, Yasuhiro Watashiba, Kohei Ichikawa, Susumu Date, Hajimu Iida, et al., Container rebalancing: Towards proactive linux containers placement optimization in a data center, in: Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual, Vol. 1, IEEE, 2017, pp. 788–795.

[89] Joel Nider, Mike Rapoport, Cross-isa container migration, in: Proceedings of the 9th ACM International on Systems and Storage Conference, ACM, 2016, p. 24.

[90] Chao-Tung Yang, Jung-Chun Liu, Shuo-Tsung Chen, Kuan-Lung Huang, Virtual machine management system based on the power saving algorithm in cloud, J. Netw. Comput. Appl. 80 (2017) 165–180.

[91] Sareh Fotuhi Piraghaj, Energy-Efficient Management of Resources in Enterprise and Container-Based Clouds (Ph.D.), University of Melbourne, Melbourne, Australia, 2016.

[92] Sareh Fotuhi Piraghaj, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, Rajkumar Buyya, A framework and algorithm for energy efficient container consolidation in cloud data centers, in: Data Science and Data Intensive Systems (DSDIS), 2015 IEEE International Conference on, IEEE, 2015, pp. 368–375.

[93] Sareh Fotuhi Piraghaj, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, Rajkumar Buyya, Containercloudsim: An environment for modeling and simulation of containers in cloud data centers, Softw. - Pract. Exp. 47 (4) (2017) 505–521.

[94] Yong Li, Wei Gao, Code offload with least context migration in the mobile cloud, in: Computer Communications (INFOCOM), 2015 IEEE Conference on, IEEE, 2015, pp. 1876–1884.

[95] Lele Ma, Shanhe Yi, Qun Li, Efficient service handoff across edge servers via docker container migration, in: Proceedings of the Second ACM/IEEE Symposium on Edge Computing, ACM, 2017, p. 11.

[96] Andrew Machen, Shiqiang Wang, Kin K. Leung, Bong Jun Ko, Theodoros Salonidis, Live service migration in mobile edge clouds, IEEE Wirel. Commun. 25 (1) (2018) 140–147.

[97] Andrew Machen, Shiqiang Wang, Kin K. Leung, Bong Jun Ko, Theodoros Salonidis, Migrating running applications across mobile edge clouds: poster, in: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, ACM, 2016, pp. 435–436.

[98] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, Khaled B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. Tutor. (2017).

[99] Chenying Yu, Fei Huan, Live migration of docker containers through logging and replay, in: Advances in Computer Science Research, International Conference on Mechatronics and Industrial Informatics, 2015.

[100] Shripad Nadgowda, Sahil Suneja, Nilton Bila, Canturk Isci, Voyager: Complete container state migration, in: Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, IEEE, 2017, pp. 2137–2142.

[101] Muhammad Zakarya, Lee Gillam, Energy and Performance Aware Resource Management in Heterogeneous Cloud Datacenters (Ph.D. thesis), University of Surrey, 2017.

[102] Sun-Yuan Hsieh, Cheng-Sheng Liu, Rajkumar Buyya, Albert Y. Zomaya, Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers, J. Parallel Distrib. Comput. (2020).

[103] Petter Svärd, Benoit Hudzia, Steve Walsh, Johan Tordsson, Erik Elmroth, Principles and performance characteristics of algorithms for live vm migration, ACM SIGOPS Oper. Syst. Rev. 49 (1) (2015) 142–155.

[104] Gang Sun, Dan Liao, Vishal Anand, Dongcheng Zhao, Hongfang Yu, A new technique for efficient live migration of multiple virtual machines, Future Gener. Comput. Syst. 55 (2016) 74–86.

[105] Petter Svärd, Benoit Hudzia, Johan Tordsson, Erik Elmroth, Evaluation of delta compression techniques for efficient live migration of large virtual machines, in: Proceedings of the 7th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, 2011, pp. 111–120.

[106] Anita Choudhary, Mahesh Chandra Govil, Girdhari Singh, Lalit K. Awasthi, Emmanuel S. Pilli, Divya Kapil, A critical survey of live virtual machine migration techniques, J. Cloud Comput. 6 (1) (2017) 23.

[107] Pokhrel Niroj, Live Container Migration: Opportunities and Challenges, Aalto University, 2017.

[108] Prateek Sharma, Lucas Chaufournier, Prashant J. Shenoy, Y.C. Tay, Containers and virtual machines at scale: A comparative study, in: Middleware, 2016, p. 1.

[109] Pavel Mach, Zdenek Becvar, Mobile edge computing: A survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. (2017).

[110] Charalampos Gavriil Kominos, Nicolas Seyvet, Konstantinos Vandikas, Bare-metal, virtual machines and containers in openstack, in: Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conference on, IEEE, 2017, pp. 36–43.

[111] Y.C. Tay, Kumar Gaurav, Pavan Karkun, A performance comparison of containers and virtual machines in workload migration context, in: Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on, IEEE, 2017, pp. 61–66.

[112] Sébastien Vaucher, Comparing virtual machines and linux containers, 2015.

[113] Alain Tchana, Noel De Palma, Ibrahim Safieddine, Daniel Hagimont, Software consolidation as an efficient energy and cost saving solution, Future Gener. Comput. Syst. 58 (2016) 1–12.

[114] Alain Tchana, Noel De Palma, Ibrahim Safieddine, Daniel Hagimont, Bruno Diot, Nicolas Vuillerme, Software consolidation as an efficient energy and cost saving solution for a saas/paas cloud model, in: European Conference on Parallel Processing, Springer, 2015, pp. 305–316.

[115] Eun Kyung Lee, Indraneel Kulkarni, Dario Pompili, Manish Parashar, Proactive thermal management in green datacenters, J. Supercomput. 60 (2) (2012) 165–195.

[116] Ali Hammadi, Lotfi Mhamdi, A survey on architectures and energy efficiency in data center networks, Comput. Commun. 40 (2014) 1–21.

[117] Daniel Guimaraes do Lago, Edmundo R.M. Madeira, Luiz Fernando Bittencourt, Power-aware virtual machine scheduling on clouds using active cooling control and dvfs, in: Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science, ACM, 2011, p. 2.

[118] US Energy Information Administration, What is us electricity generation by energy source? 2019.

[119] Chao Li, Rui Wang, Tao Li, Depei Qian, Jingling Yuan, Managing green datacenters powered by hybrid renewable energy systems, in: 11th International Conference on Autonomic Computing, {ICAC} 14, 2014, pp. 261–272.

[120] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, Niangjun Chen, Data center demand response: Avoiding the coincident peak via workload shifting and local generation, Perform. Eval. 70 (10) (2013) 770–791.

[121] Enida Sheme, Patricia Stolf, Georges Da Costa, Jean-Marc Pierson, Neki Frashëri, Efficient energy sources scheduling in green powered datacenters: A cloudsim implementation, 2016.

[122] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, Chris Hyser, Renewable and cooling aware workload management for sustainable data centers, in: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, 2012, pp. 175–186.

[123] Joseph Doyle, Robert Shorten, Donal O'Mahony, Stratus: Load balancing the cloud for carbon emissions control, IEEE Trans. Cloud Comput. 1 (1) (2013) 1.

[124] Íñigo Goiri, William Katsak, Kien Le, Thu D. Nguyen, Ricardo Bianchini, Parasol and greenswitch: Managing datacenters powered by renewable energy, ACM SIGPLAN Not. 48 (4) (2013) 51–64.

[125] Minxian Xu, Rajkumar Buyya, Managing renewable energy and carbon footprint in multi-cloud computing environments, J. Parallel Distrib. Comput. 135 (2020) 191–202.

[126] Xiang Deng, Di Wu, Junfeng Shen, Jian He, Eco-aware online power management and load scheduling for green cloud datacenters, IEEE Syst. J. 10 (1) (2014) 78–87.

[127] Kashif Bilal, Saif Ur Rehman Malik, Osman Khalid, Abdul Hameed, Enrique Alvarez, Vidura Wijaysekara, Rizwana Irfan, Sarjan Shrestha, Debjyoti Dwivedy, Mazhar Ali, Usman Shahid Khan, Assad Abbas, Nauman Jalil, Samee U. Khan, A taxonomy and survey on Green Data Center Networks, Future Gener. Comput. Syst. 36 (2014) 189–208.

[128] Steven Lanzisera, Bruce Nordman, Richard E. Brown, Data network equipment energy use and savings potential in buildings, Energy Efficiency 5 (2) (2012) 149–162.

[129] Raffaele Bolla, Franco Davoli, Roberto Bruschi, Ken Christensen, Flavio Cucchietti, Suresh Singh, The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization? IEEE Commun. Mag. 49 (8) (2011) 80–86.

[130] Dzmitry Kliazovich, Pascal Bouvry, Yury Audzevich, Samee Ullah Khan, Greencloud: a packet-level simulator of energy-aware cloud computing data centers, in: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, IEEE, 2010, pp. 1–5.

[131] Luiz André Barroso, Urs Hölzle, Parthasarathy Ranganathan, The datacenter as a computer: Designing warehouse-scale machines, Synth. Lect. Comput. Archit. 13 (3) (2018) i–189.

[132] Víctor Medel, Rafael Tolosana-Calasanz, José Ángel Bañares, Unai Arronategui, Omer F. Rana, Characterising resource management performance in kubernetes, Comput. Electr. Eng. 68 (2018) 286–297.

[133] H P technical white paper, Linux container performance on hpe proliant servers: Understanding performance differences between containers and virtual machines, 2016.

[134] Yang Hu, Huan Zhou, Cees de Laat, Zhiming Zhao, Concurrent container scheduling on heterogeneous clusters with multi-resource constraints, Future Gener. Comput. Syst. 102 (2020) 562–573.

[135] K. Kaur, S. Garg, G. Kaddoum, S.H. Ahmed, M. Atiquzzaman, Keids: Kubernetes-based energy and interference driven scheduler for industrial iot in edge-cloud ecosystem, IEEE Internet Things J. 7 (5) (2020) 4228–4237.

[136] Liang Lv, Yuchao Zhang, Yusen Li, Ke Xu, Dan Wang, Wendong Wang, Minghui Li, Xuan Cao, Qingqing Liang, Communication-aware container placement and reassignment in large-scale internet data centers, IEEE J. Sel. Areas Commun. 37 (3) (2019) 540–555.

[137] Yuqi Fu, Shaolun Zhang, Jose Terrero, Ying Mao, Guangya Liu, Sheng Li, Dingwen Tao, Progress-based container scheduling for short-lived applications in a kubernetes cluster, in: 2019 IEEE International Conference on Big Data, Big Data, IEEE, 2019, pp. 278–287.

[138] MinSu Chae, HwaMin Lee, Kiyeol Lee, A performance comparison of linux containers and virtual machines using docker and kvm, Cluster Comput. 22 (1) (2019) 1765–1775.

[139] Fan Jiang, Kyle Ferriter, Claris Castillo, A cloud-agnostic framework to enable cost-aware scheduling of applications in a multi-cloud environment, in: NOMS 2020 - IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, April, 2020, 20–24, IEEE, 2020, pp. 1–9.

[140] Isabelly Rocha, Christian Göttel, Pascal Felber, Marcelo Pasin, Romain Rouvoy, Valerio Schiavoni, Heats: Heterogeneity-and energy-aware task-based scheduling, in: 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP, IEEE, 2019, pp. 400–405.

[141] Neeraj Kumar, Gagangeet Singh Aujla, Sahil Garg, Kuljeet Kaur, Rajiv Ranjan, Saurabh Kumar Garg, Renewable energy-based multi-indexed job classification and container management scheme for sustainability of cloud data centers, IEEE Trans. Ind. Inf. 15 (5) (2018) 2947–2957.

[142] Yang Hu, Cees De Laat, Zhiming Zhao, et al., Multi-objective container deployment on heterogeneous clusters, in: Proc. 19th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., CCGRID, 2019, pp. 592–599.

[143] Mainak Adhikari, Satish Narayana Srirama, Multi-objective accelerated particle swarm optimization with a container-based scheduling for internet-of-things in cloud environment, J. Netw. Comput. Appl. 137 (2019) 35–61.

[144] Kuljeet Kaur, Sahil Garg, Gagangeet Singh Aujla, Neeraj Kumar, Albert Zomaya, A multi-objective optimization scheme for job scheduling in sustainable cloud data centers, IEEE Trans. Cloud Comput. (2019).

[145] Sukhpal Singh Gill, Peter Garraghan, Vlado Stankovski, Giuliano Casale, Ruppa K. Thulasiram, Soumya K. Ghosh, Kotagiri Ramamohanarao, Rajkumar Buyya, Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge, J. Syst. Softw. 155 (2019) 104–129.

[146] M. Zakarya, L. Gillam, H. Ali, I. Rahman, K. Salah, R. Khan, O. Rana, R. Buyya, epcaware: A game-based, energy, performance and cost efficient resource management technique for multi-access edge computing, IEEE Trans. Serv. Comput. (2020) 1–14.

[147] Mar Callau-Zori, Lavinia Samoila, Anne-Cécile Orgerie, Guillaume Pierre, An experiment-driven energy consumption model for virtual machine management systems, Sustain. Comput.: Inform. Syst. 18 (2018) 163–174.

[148] Muhammad Zakarya, Lee Gillam, Ayaz Ali Khan, Izaz Ur Rahman, Perficientcloudsim: a tool to simulate large-scale computation in heterogeneous clouds, J. Supercomput. (2020) 1–55.

[149] Haikun Liu, Hai Jin, Cheng-Zhong Xu, Xiaofei Liao, Performance and energy modeling for live migration of virtual machines, Cluster Comput. 16 (2) (2013) 249–264.

[150] Ayaz Ali Khan, Muhammad Zakarya, Rajkumar Buyya, Rahim Khan, Mukhtaj Khan, Omer Rana, An energy and performance aware consolidation technique for containerized datacenters, IEEE Trans. Cloud Comput. (2019).

[151] Minxian Xu, Rajkumar Buyya, Brownoutcon: A software system based on brownout and containers for energy-efficient cloud computing, J. Syst. Softw. 155 (2019) 91–103.

[152] Changyeon Jo, Youngsu Cho, Bernhard Egger, A machine learning approach to live migration modeling, in: Proceedings of the 2017 Symposium on Cloud Computing, ACM, 2017, pp. 351–364.

[153] Gueyoung Jung, Matti A. Hiltunen, Kaustubh R. Joshi, Richard D. Schlichting, Calton Pu, Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures, in: Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, IEEE, 2010, pp. 62–73.

[154] Adel Nadjaran Toosi, Chenhao Qu, Marcos Dias de Assunção, Rajkumar Buyya, Renewable-aware geographical load balancing of web applications for sustainable data centers, J. Netw. Comput. Appl. 83 (2017) 155–168.

[155] Shreshth Tuli, Nipam Basumatary, Sukhpal Singh Gill, Mohsen Kahani, Rajesh Chand Arya, Gurpreet Singh Wander, Rajkumar Buyya, Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments, Future Gener. Comput. Syst. 104 (2020) 187–200.

[156] Muhammad Ali, Ashiq Anjum, M. Usman Yaseen, A. Reza Zamani, Daniel Balouek-Thomert, Omer Rana, Manish Parashar, Edge enhanced deep learning system for large-scale video stream analytics, in: Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on, IEEE, 2018, pp. 1–10.

[157] Redowan Mahmud, Satish Narayana Srirama, Kotagiri Ramamohanarao, Rajkumar Buyya, Profit-aware application placement for integrated fog–cloud computing environments, J. Parallel Distrib. Comput. 135 (2020) 177–190.

[158] Sharrukh Zaman, Daniel Grosu, A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds, IEEE Trans. Cloud Comput. 1 (2) (2013) 129–141.

[159] José Moura, David Hutchison, Game theory for multi-access edge computing: Survey, use cases, and future trends, IEEE Commun. Surv. Tutor. 21 (1) (2018) 260–288.

[160] Qiang He, Guangming Cui, Xuyun Zhang, Feifei Chen, Shuiguang Deng, Hai Jin, Yanhui Li, Yun Yang, A game-theoretical approach for user allocation in edge computing environment, IEEE Trans. Parallel Distrib. Syst. 31 (3) (2020) 515–529.

Ayaz Ali Khan is currently a Ph.D. student in the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. He completed his M.Phil (MS) in computer science from COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan. His area of research includes energy-aware and performance-efficient scheduling, resource allocation, placement and management, at datacenter level. Moreover, he has enough knowledge of distributed systems, optimization, game theory and computer programming.

Muhammad Zakarya is a Senior Member of the IEEE. He received the Ph.D. degree in computer science from the University of Surrey, Guildford, U.K. He is currently a Lecturer with the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. His research interests include cloud computing, mobile edge clouds, performance, energy efficiency, algorithms, and resource management. He has deep understanding of the theoretical computer science and data analysis. Furthermore, he also owns deep understanding of various statistical techniques which are, largely, used in applied research. His research has appeared in highly ranked IEEE transactions, journals and conferences. He serves as reviewer for more than 20 international journals and conferences. He is an Associate Editor for the IEEE Access Journal and has served as TPC member in various international level conferences and workshops.