Wireless Personal Communications (2020) 115:2289–2326
https://doi.org/10.1007/s11277-020-07682-8

Virtual Machine Consolidation in Cloud Computing Systems: Challenges and Future Trends

Rahmat Zolfaghari[1,2] · **Amir Masoud Rahmani**[3]

Published online: 8 August 2020

Abstract
Cloud Computing Systems (CCSs) provides a computing capability through the Internet. It enables organizations or individuals to have a computing power without deploying and maintaining their own Information Technology infrastructure. As a cloud is realized on a vast scale cloud, it consumes an enormous amount of energy. Migration pattern, where several Virtual Machines (VMs) can be placed on a minimum number of active Physical Machines is called VMs Consolidation (VMC). Thus, this technique can be a practical approach for balancing electricity consumption and other QoS requirement in CCSs. Especially, VMC must meet the service quality requirements, minimization of both energy consumption and Service Level Agreement violation in CCSs. This paper presents a systematic survey of VMC in CCSs with particular attention to the VMC phases, metrics, objectives, migration patterns, optimization methods, and evaluation approaches of VMC. Our review study is presented based on the past literature with a focus on the type of hardware metrics, software metrics, objectives, algorithms, and architectures of VMC in CCSs.

Keywords Cloud Computing Systems (CCSs) · Data center · VMs Consolidation (VMC) · Energy consumption · QoS · Systematic survey

1 Introduction

CCSs consume a massive amount of electricity resulting in high CO_2 emission [1]. For optimizing resource utilization in CCSs and reducing electricity consumption, VMC can be used by switching the idle PMs to silent or sleep mode [2–4]. VMC approaches consolidate several VMs onto a PM or fewer numbers of PMs to reduce electricity consumption and optimize resource utilization. This occurs by supporting the PMs run optimally on efficient electricity and more electricity proportional conditions [5]. The main feature that makes the VMC techniques more important is migration, especially live

✉ Amir Masoud Rahmani
arahmani@khazar.org; rahmani74@yahoo.com

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Department of Computer Engineering, Hashtgerd Branch, Islamic Azad University, Alborz, Iran

[3] Department of Computer Science, Khazar University, Baku, Azerbaijan

VMs migration. Indeed, live VMs migration can transfer a running VM from a PM to another PM with no interruption in services [6].

VMC can be done in different ways, considering criteria, resources, objectives, and algorithmic methods [7, 8]. Due to the importance of VMC, some research has been conducted to address reducing energy consumption in CCSs.

To meet the growing needs of large databases and computational resources, data centers have to use high-performance PMs as well as vast and high-speed resources of PMs. These resources, along with cooling equipment and network alternatives, are among the largest power consumers in the cloud [9]. This power consumption in CCSs is proportional to the number of resources, and also it is one of the highest power consumers in the world [10]. Further, according to the Gartner report in 2013, on average, the energy consumed by a data center is equivalent to the energy consumption of 25,000 families [11]. In another report in 2014, would be it was stated that the federal and cloud energy consumption in the US was about 100 billion kW/h and the cost of energy about 75% of the total operating costs [12]. In the USA is predicted to consume 140 billion kilowatts per year by 2020 and resulting in $13 billion per year in electricity bills for cloud [13]. Also, the statistics indicate a growth in energy consumption in CCSs, and overall energy consumption incurs enormous costs; it has to be reduced [14–16]. According to studies, the low utilization of resources in the data center and cloud also increases power consumption. The average utilization of data centers is very low, and between 12 and 18% [17]. Cloud has a better status in comparison to the regular data centers, whose utilization is claimed to be 40–70% [18]. So, energy consumption in CCSs is a challenge, and the VMC technique is its solution.

In addition to the energy in VMC, other metrics such as reliability, the hardware cost, and its longevity, performance, ON–OFF power cycles, cooling, load balancing, SLAv, VMs affinity, Network Band Width (NBW), resource utilization, etc. can also affect the VMC technique. So, these metrics, along with reducing energy consumption in VMC technique, have been considered in some studies as follows.

Meanwhile, less attention has been paid to reliability reduction and increased server fatigue due to the over-aggressive domination of CCSs. Therefore, the important thing is to use methods that emphasize short-term energy consumption, saving policies, and paying attention to the long-term costs such as the hardware cost incurred on CCSs due to diminished PMs reliability by increasing ON–OFF cycles and depreciation of PMs, the temperature rise and thermal [19].

It is claimed that repeatedly turning PMs ON–OFF can lead to hardware failures. Note that when PM fail because of hardware errors, they would be redundant as soon as they start to work again, which might cause massive damage to CCSs given the effort needed for repairing and replacing these PMs [20]. Hardware manufacturing companies have typically cited that they can only guarantee the performance and reliability in case of at most 60,000 ON–OFFs throughout their whole life cycle [21].

Further, the temperature and thermal of the cloud are other factors that cause PMs to fail. Temperature and thermal are one of the most influential factors in the reliability of efficiency, which can affect the longevity and power consumption for cooling of PMs in CCSs and is considered to be very helpful in resource allocation [22]. Upon elevation of the PMs' temperature of by 10 °C, the longevity of PMs' components is reduced by 50%. These failures can result in a partial or total failure of one or more PMs in CCSs, which can incur a penalty of 5000$ per minute to cloud as SLAv cost [23].

As well, reducing power consumption along with load balancing in [24, 25], in [26] reducing power consumption along with performance, in [27] reducing power consumption

along with NBW, and in [28, 29] reducing power consumption along with VMs affinity to implement the VMC technique in CCSs considered.

In summary, due to high rates of energy consumption and low utilization rate in the cloud, the importance of VMC technique as its solution is clear. So far, various metrics and objectives have been proposed to implement the VMC technique in CCSs. The best way to placement VMs to PMs is not merely inserting the maximum number of VMs into the minimum number of PMs. Because in this case, in addition to the energy consumption resulting in high costs and CO_2 emissions to the environment, important criteria must be considered in VMC approaches such as migration overhead [30–32], performance [7, 33–35], SLAv [23, 36], cooling [5, 37, 38], thermal and temperature [21, 30, 39], ON–OFF cycles [22, 40], VMs affinity [7, 28, 29, 41], reliability [19, 20], the hardware cost and its longevity [22, 23], load balancing [24, 25], NBW [27, 42–44], resources utilization [8, 45, 46]. In other words, to implement the VMC algorithms, reducing power consumption along with the mentioned criteria, must be considered for holistic efficiency in the cloud.

Finally, the implementation problem of VMC is generally NP-hard [28], and different methods have been proposed to its solution. Some of them are based on exact methods, such as; linear Programming [47, 48], dynamic programming [5, 15], and Constraint Satisfaction Problems (CSP) [35, 49]. The problem of bin packing as greedy methods such as; First Fit Decreasing (FFD) [50], Best Fit Decreasing (BFD) [34], Modify Best Fit Decreasing (MBFD) [51, 52]. Evolutionary methods such as the Genetic Algorithm (GA) [53], Ant Colony Optimization (ACO) [54], Particle Swarm Optimization (PSO) [55], etc. Which several studies with these algorithms to implement the VMC in CCSs presented in Sect. 2.

Our contributions to this work are as following:

1. Presenting a categorized overview of the criteria and objectives to VMC, which, highlighted these criteria used to evaluate existing literature.
2. Presenting a categorized overview of the algorithms to VMC and methods for VMC evaluation, which, highlighted their benefits and weaknesses.
3. Presenting challenges, open issues, and suggested future works.

In the following, in Sect. 2 of this paper, in several studies, the algorithms to implement the VMC in CCSs presented. The studied algorithms are of three types; exact methods, greedy methods, and evolutionary methods along with a variety of hardware and software criteria, in most of the studies, a tradeoff between energy consumption and QoS criteria in CCSs has been considered.

2 Related Works

In this section, the authors have presented comparative analysis and description of energy-aware resource allocation algorithms and techniques for VMC in CCSs, which are used for making more energy-efficient VMs. Cloud computing has gained a wide range of attention in both industry and academics as cloud services offer a pay-per-use model, due to the increase in need of factors like reliability and computing results with immense growth in cloud-based companies along with a continuous expansion of their scale. However, the rise in cloud computing users can cause a negative impact on energy consumption in the CCSs as they consume a massive amount of overall energy. In order to minimize energy

consumption in the CCSs, researchers proposed various energy-efficient resources management strategies. Dynamic VMC is one of the prominent techniques and an active research area in recent times, used to improve resource utilization and minimize the energy consumption of CCSs. This technique monitors the CCSs utilization, identify overloaded, and underloaded PMs then migrate few/all VMs to other suitable PMs using VMs selection and VMs placement, and switch underloaded PMs to sleep mode.

This section discusses some papers that refer to the VMC; different methods and algorithms have been proposed to solve it. In this section, a survey in several studies that provided the algorithms based on considered QoS metrics for the VMC is presented. These algorithms for implementation of VMC are divided into static and dynamic; however, dynamic VMC is more used. The studied algorithms are categorized into three types: exact methods, greedy methods, and evolutionary methods, along with a variety of hardware and software metrics. VMC algorithms have been considered in several studies as follows, which, in this studies reducing energy consumption is the main objective, but other metrics along with it must be considered. In other words, for overall efficiency in CCSs, it must be considered a tradeoff between metrics/objectives in VMC algorithms. In most of the studies, a tradeoff between energy consumption and other QoS metrics in CCSs has been considered.

The authors [5] presented an exact method as an Integer Linear Programming (ILP)—based VMs placement method to find the best location of each PM in the CCSs based on its power consumption. The authors provided a way to reduce power consumption in the CCSs. Via this method, they tried to reduce the rotational airflow in cloud corridors considering the number of active PMs, the number of active shelves, and effect active PMs have on each other. The authors [7] presented the efficiency limitation of any given PM with VMs on it. If this limitation is met, it can lead to elevated resource efficiency, as PMs may not use enough resources to ensure that there is no kind of performance degradation. Thus, at first, this method obtains the extent of the negative impacts of VMs on each other, after which VMs are mounted on a PM. Accordingly, affinitive VMs and VMs that made each other's performance least efficient were placed on a PM. Therefore, in addition to reducing energy consumption by VMC, the service could also be enhanced. In this research, to considering of VMs affinity to implement the VMC technique in CCSs emphasized. In [20], an exact algorithm called the "Markov model" was introduced, which established a tradeoff between three parameters, including performance, cost, and reliability. Initially, this algorithm predicted the number of resources for the future and then turned on servers to boost performance even before they needed it. This algorithm also maximized power saving while minimizing the need for unresponsive sources. Also, to increase the reliability, the impact of PM' ON–OFF was considered in the algorithm. In [28], a greedy algorithm based on the problem of constraint satisfaction for VMC was proposed. This article aimed to reduce the number of active PMs required and the number of migrations; But in this method, authors did not consider SLAv and the number of migrations metrics. A distributed Dynamic Virtual Machine Consolidation (DVMC) was introduced in [42], which, DVMC method, is the process of reducing the number of active PMs through live VMs migration to diminish energy consumption and improve resource utilization of PMs in CCSs. As well, this method cause diminishes energy consumption along with minimizing the number of VMs migrations and SLAv in CCSs. In [43], the location of VMs was chosen to reduce the amount of intra-cloud data center traffic as well as inter-cloud data center traffic along with reducing energy consumption in VMC. Clearly, with- increasing the distance inter VMs, the available bandwidth among them will decrease while the delay time of the applications will increase; thus, this might lead to service efficiency reduction.

A distributed evolutionary method, Ant Colony Optimization System (ACOS) was introduced in [33]. In this research, a new algorithm based on ACOS to solve the VMC problem aims to save the energy consumption of CCSs. It significantly reduces the number of migrations and the active PMs that result in the reduction of total energy consumption of CCSs. In [56], the authors, an exact method as Energy and Thermal-Aware Scheduling (ETAS) algorithm that dynamically consolidates VMs to minimize the overall energy consumption while proactively preventing hotspots. ETAS is designed to address the trade-off between energy, hotspots, and SLAv. In [48], an exact method was proposed based on linear programming, which, the authors compared their method with different greedy methods and claimed that if they do not displace VMs whose necessary resources have not been changed, the number of migrations will be reduced. However, the number of required PMs will not change considerably. This algorithm was executed with and without controlling migration, and the results indicated that the number of migrations was reduced by this method, but the number of PMs did not show much increase. In [52], a centralized method was proposed based on greedy methods to solve the problem of VMC. They used the MBFD algorithm to determine the PM of VMs. This algorithm improvement in energy consumption and reduction in SLAv, the number of active PMs, and the amount of VMs migration. In [57], the main idea was to optimize the network and the relationship between VMs. The cost of the relationship between VMs was modeled as the product of the delay in the relation rate between the two VMs. In this method, only the network was considered, and SLA might have been violated because of not taking other resources such as the processor into account. Decreasing migration overhead leads to QoS. In [53], a dynamic evolutionary method was introduced based on the GA has been proposed for the VMC problem is used to reduce the search space of the VMC problem. The size of the search space of the VMC problem is determined by the number of PMs and VMs that need to be considered. This method, reducing the number of PMs and the number of VMs contributes to the reduction of the number of VM migration. Also, an effective method for reducing power consumption. In [54] an evolutionary distributed algorithm based on the ACO algorithm called "ACO cloud" was introduced. In the method of the ACO cloud, there are lower and upper thresholds for the utilization of the resources. Every PM views the processor and memory utilization. When utilization of a resource becomes less than the lower threshold or exceeds the upper threshold, they should decide for migration. This decision-making is done by a Bernoulli trial. In case of having a successful Bernoulli trial, one of the VMs is selected for migration, and the request for migration is sent to the other PMs. PMs also perform a Bernoulli trial for accepting the VM. The probability of the success of Bernoulli trials is dependent on the utilization of the resources, and changes due to its fluctuation. In the simulation, utilization of resources, energy consumption, and SLAv are considered as a metric. In [58], the authors an algorithm based on the greedy method provided. They used the FF and MBFD greedy algorithms to optimized VM to PM mapping. The authors propose a novel and effective greedy approach for VM allocation that can maximize the energy efficiency of the cloud. This approach can consolidate more VMs with fewer PMs to achieve better energy efficiency than popular methods. In [59], the authors introduced an evolutionary method, named VMs Placement Biogeography-Based Optimization (VMPBBO). Their method constituted a sampling of migration and habitat of living creatures in the islands of the ocean. Each island can be the habitat for living creatures. Habitat suitability index is measured by some parameters called Suitability Index Variables (SIV). These parameters reflect some variables, such as temperature, along with the weather condition on each island. Over time, these parameters changed and made living creatures migrate among the islands. These islands are within some archipelagoes.

Migration within the archipelago has low costs, while migrating between archipelagoes is costlier. In the algorithm of VMPBBO, each archipelago corresponds to a cluster, and each island corresponds to a PM. In the presented algorithm, VMs are positioned on the PMs with a greater habitation capability. In [60], a distributed method named V-MAN was introduced, in which each VM sends its utilization of resources to other PMs. All the PMs that receive the information have three options regarding their utilization of resources. The first option is to receive all the VMs positioned on the PM so that the machine will turn off. The second option is that it sends all of its VMs to the PM so that it turns off. The third option is not to participate in migration. In this method, the only resource considered is the CPU, while other resources, SLAv, and the number of migrations are not considered. In [29], the authors tried to determine VMs to PMs by considering the migration overhead, so that migrations would have the minimum negative effect on its QoS. For example, because of the similarity in memory pages of VMs, all VMs with similar pages would be placed on the same PM. In [61], a greedy method was offered for placing VMs with common memory pages on a PM. The similarity between these programs can also lead to an algorithm trying to place VMs with similar practical programs in a PM. In [62], linear regression was used for predicting the utilization of the resources of VMs. The authors indicated in their experiments that VMs could be positioned on PMs via prediction. The results revealed that by using prediction, both energy consumption and SLAv would diminish. In [63], the authors proposed an evolutionary algorithm named Grey Wolf Optimization (GWO) for VMs Placement (VMP) phase of VMC. This method reduces the number of active PMs, energy consumption, SLAv, the number of migration, and the more efficient use of CPU and RAM resources. In [50], the authors introduced the thresholds policy for the utilization of resources, where the algorithms utilized the resources to lie between these two threshold ranges. The upper thresholds prevent the SLAv, while the lower ones prevent PMs from becoming idle. In [64], an algorithm based on threshold policy and an evolutionary algorithm as ACOS was introduced. Which, leverages (lower/upper) thresholds of CPU utilization to identify the PM load status, VMC is triggered when the PM is overloaded or under-loaded. During VMC, the approach selects migration VMs and destination PMs simultaneously based on ACOS, utilizing various selection policies according to the PM load status. This method causes more optimization energy consumption, SLAv, performance, the number of migration, and the more efficient use of processor and memory resources. In [65], an evolutionary algorithm based on a GA named Improved Grouping Genetic Algorithm for VMC (IG2CA) in CCSs proposed. That it considers the metrics comprising hardware longevity, reliability, and power reduction in CCSs. This algorithm uses these parameters to find the best way to map from VMs on PMs. Finally, the mapping will be selected to have an optimal sum of these three parameters.

As well, in several studies on VMC in CCSs, reducing energy consumption along with other essential metrics in VMC considered. Such as; in [66] a fast evolutionary algorithm as Simulated Annealing based Resource Consolidation algorithm (SARC) for reducing energy consumption along with optimizes resource utilization in cloud comprising a memory, processor, and NBW was presented. In [56], a greedy method, using constraint-based multi-objective optimization, for reducing energy consumption along with optimizes SLAv, performance, and the more efficient use of CPU and RAM resources provided. In [39], a dynamic method, namely Energy and Thermal-Aware Scheduling (ETAS), for reducing energy consumption along with reducing SLAv and thermal, boost reliability and performance provided. In [67], the authors presented an exact method as Mixed-Integer Linear Programming (MILP) for reducing energy consumption along with optimizing performance. In [36], reducing energy consumption along with optimizes SLAv and performance

presented. In [55], an evolutionary method as the Modified Particle Swarm Optimization (MPSO) method for reducing energy consumption along with optimizes performance metrics presented. As well, the resource utilization of PMs comprising processor and storage was considered. In [68], the authors presented an algorithm as the ILP algorithm for reducing energy consumption along with optimizes SLAv and performance. In [69], an exact algorithm as the MILP algorithm for reducing energy consumption along with reducing SLAv, the number of migration, and the more efficient use of CPU resources provided.

In this research, contrary to the surveyed studies, a systematic review is presented with a category of different algorithms include: (exact method, heuristic methods, meta-heuristic methods), metrics and objective functions include: (energy consumption, efficiency, SLA violation, the operational and hardware cost, thermal, ON–OFF power cycles, hardware and resource utilization, NBW, cooling systems, migration execution time, VMs affinity, load balancing, the performance of services, the reliability of CCSs devices, and migration overhead) are considered on VMs consolidation methods. Also, contrary to the studied papers, an association of utilization rate of hardware resources (CPU, RAM, NBW, DISK) as hardware metrics and other important VMs consolidation metrics include: (migration overhead, SLA violation, VMs affinity, load balancing) as software metrics are considered in VMs consolidation methods.

The rest of this research is organized as follows: Basic concepts are presented in Sect. 3. Section 4 presents the selection process based on a systematic survey. A VMs consolidation taxonomy is discussed in Sect. 5. Challenges and technical issues in VMC are presented in Sect. 6. Finally, the conclusion and future work are presented in Sect. 7.

3 Basic Concepts

In this section, some essential items, such as virtualization, VMC, SLA, and VMs migration, which is necessary for this study, are briefly presented.

3.1 Virtualization

Virtualization has turned out to be a key asset in many areas of information technology. Cloud computing is generally based on the concept called virtualization. In computing, virtualization refers to the creation of a virtual version to any device or resource, such as a server, operating system, storage device, or network, which means the framework divides a resource into multiple execution environments. Virtualization plays an essential role in organizing and managing access to the pool of resources employing a software layer termed as a Virtual Machine Monitor (VMM) or hypervisor. For higher-level applications, it hides the details of the physical resources and only provides virtualized resources.

Furthermore, it virtualizes the entire resources of a given PMs, allowing several VMs in it to share its resources. Microsoft hyper-v, Xen, ESX, oracle virtual box, and kernel-based KVM, are some of the popular virtualization software. One main advantage of virtualization is that it allows the opportunity of assigning multiple VMs into a single PM using a technique referred to VMC. It also provides a capability called VM live migration, i.e., the ability to relocate a VM from one PM to the other PM, nearly with a zero downtime. The hypervisor is a software layer between the PM and Operating System. It avoids application privacy violations by running a few VMs on PM in an isolated secure environment

to achieve the SLA in the cloud. In simulations, XEN and KVM are used more frequently [16].

Virtualization provides many benefits, as follows:

1. The feasibility of running multiple PMs and providing multiple services simultaneously by a PM.
2. Optimizing the efficiency of PM resources and reducing the number of active PMs in CCSs.
3. The possibility of live migration of VMs to PMs with minimum service downtime making it feasible to respond quickly to the load changes of CCSs.
4. The ability to change the number of resources assigned to VMs dynamically and without interrupting services.

3.2 VMC

The VMC section comprising two subsection background and system architecture of VMC described in detail.

3.2.1 Background of VMC

A practical VMC framework constitutes algorithms that resolve three subproblems/steps as following.

1. PMs Detection: a decision when to start a VM migration;
2. VMs Selection: a selection of which VMs to migrate;
3. VMs Placement: a selection of PMs for placement;

that is explained in the following. The VMC steps Shown in Fig. 1.

1. PMs Detection

In this step as when to migrate VMs, the amount load of PM is detected by applying the over-loaded/under-loaded detection algorithms. Generally, all algorithms in this step are exact methods. These algorithms are comprising; Static/Dynamic Threshold (ST/DT), Median Absolute Deviation (MAD), Inter Quartile Range (IQR), Local Regression (LR), Robust Local Regression (LRR) and Markov [70]. This step is provided to reduce the power consumption, minimizing performance degradation due to migration, load balancing, minimizing SLA violation, and simultaneously fulfilling the required QoS by over-loaded/under-loaded PM detection techniques. Also, machine learning approaches such as KNN, ANN, SVM, and RF can be used for load prediction in PMs, in cases where it is necessary to predict a load of PMs in the CCSs [1].

2. VMs Selection

VMs Selection as which VM(s) to migrate is once it has been decided that a PM is overloaded, the next step is to select particular VMs to migrate from this PM. The policies for

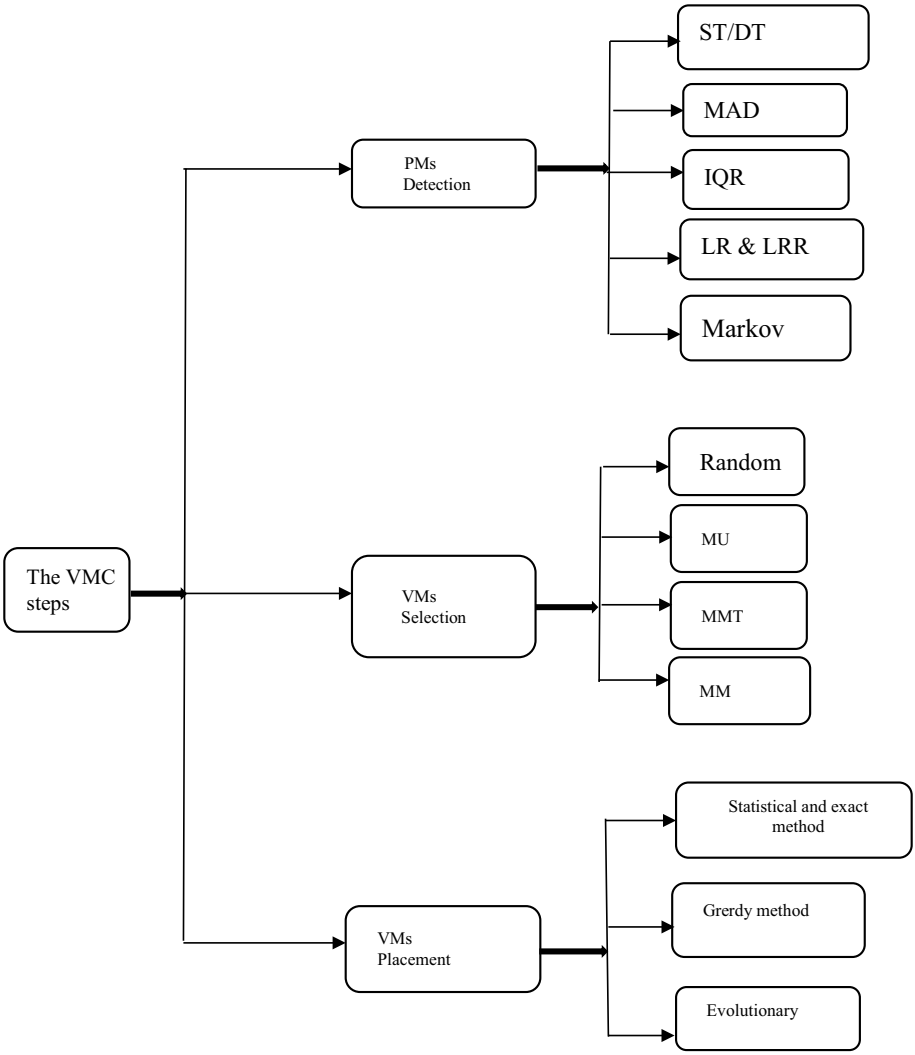

Fig. 1 The VMC steps

VM selection can be various. These policies are applied iteratively, after a selection of a VM to migrate. The PM is rechecked for being overloaded; if it is still considered as being overloaded, the VM selection policy is applied again to select another VM to migrate from the PM. This is repeated until the PM is considered as being not overloaded. So, VM selection is used to select a particular VM to migrate from PM to another PM. There are some policies for VMs selection as following [1, 61, 71]:

1. The Random Choice (RC) Policy: randomly selects the VMs to migrate.
2. The Minimum Utilization (MU) Policy: selects those VMs that have the lowest usage of CPU.

3. The Minimum Migration Time (MMT) Policy: This policy migrates a VM that requires the minimum time to complete a migration compared to other VMs.
4. The Highest Potential Growth (HPG) Policy: When the upper threshold is violated, policy migrates VMs that have the lowest usage of the CPU relative to the CPU capacity defined by the VM parameters in order to minimize the potential increase of the PMs' utilization and prevent a SLAv.
5. The Minimization of Migration(MM) Policy: The MM policy selects the minimum number of VMs needed to migrate from a PM with lower CPU utilization below the upper utilization threshold if the upper utilization threshold is violated. In this policy, only CPU utilization is considered.

3. VMs Placement

VMs placement as to where to migrate selects the target PMs at which VM has to place after migration. We choose the most power-efficient PM for VM placement with the condition that it becomes not overloaded after migration. In many previous articles, the exact method [20–23], greedy [28, 42, 43], and evolutionary [44, 51, 66] for this section of VMC have been used.

3.2.2 Architecture of VMC

The architecture of VMC is portrayed in Fig. 2 that comprises three phases as follows.

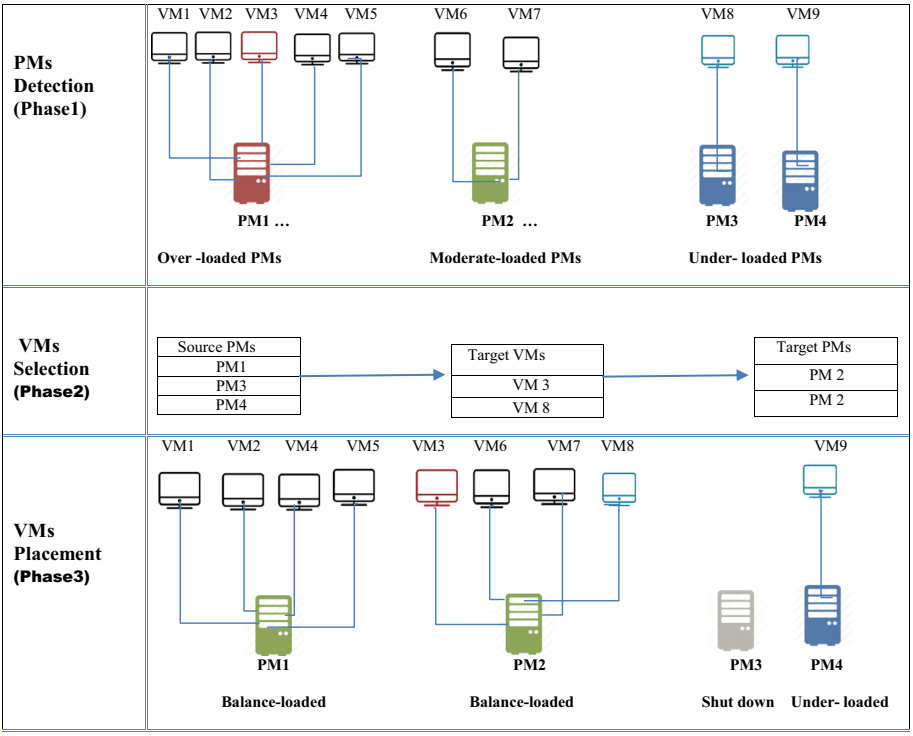

Fig. 2 The VMC architecture

1. Before VMC.
2. During VMC.
3. After VMC.
1. Before VMC.

In phase 1, as PMs Detection, based on the resources utilization level, such as CPU of PMs, the PMs Detection algorithm detects three types; over-loaded PM such as PM1, moderate-loaded PM such as PM2, and under-loaded PM such as PM3. On one side, PMs at over-loaded status can easily cause high SLA violations. On the other side, under-loaded PMs, which still need to remain alive, increase unnecessary energy costs. Therefore, if no VMC strategy is interposed to the inefficient usage of computing resources, then much extra energy will be wasted, as well as lower QoS.

2. During VMC.

In phase 2, as VMs Selection, the procedure of VMC can be comprised of three steps: step (1) Source PMs Selection such as PM1and PM3; step (2) Target VMs selection such as VM3 and VM8; step (3) Target PMs selection such as PM2. In step (1), a set of PMs is selected for VMs to migrate out. This step takes all the PMs and VMs as input and selects one or more PM(s) as the source PM named PM1, PM3. In step(2), one or more VM(s) such as VM3 and VM8 are selected from a source PMs for migration. Over-loaded VMs of Over-loaded PMs such as VM3 along with VMs of Under-loaded PMs such as VM8 migrate to Moderate-loaded PMs such as PM2.

Migration of the VMs from under-loaded PMs does not require a policy because all VMs must be migrated If the destination PM has the necessary capacity, but, for selecting a VMs from over-loaded PMs, an appropriate policy must be adopted. In step(3), a PM is selected to hold the selected VMs from step 2.

3. After VMC.

In phase C, as VMs Placement, after adding VMC strategy, there are only three PMs that are alive in the system, most of them are at balance-loaded status, which causes further decreases in SLA violation and causes high QoS. In addition, the under-loaded PMs such as PM 3 is shut down, which cause further decreases in energy consumption. As well as, utilization of resources of PMs close to optimum utilization point for most active PMs and other PMs are shut down, which are portrayed in Fig. 1. In Fig. 1, it is assumed, the destination PM hasn,t the necessary capacity for VM9, or there is no suitable PM for it, therefore not migrate out. While, if VM9 was migrated, it causes more improvement in balance degree, utilization resources, energy consumption.

Finally, most PMs are Balanced, and unloaded PMs have been turned off. Therefore, this architecture causes a tradeoff between energy consumption and other QoS requirement in CCSs.

3.2.3 Description of the Methodology of VMC

When a low workload is imposed on modern computers, their electric power is equal to a large percentage of their maximum electricity consumption. For this reason, the low utilization of data centers leads to energy waste. The maximum capacity of resources should be

used to reduce power consumption. VMs with a lower load can be dislocated to other PMs, while source PMs are turned off to reduce power consumption in CCSs. This method is called VMC methodology [16, 72]. The migration, from overload PM for the obligation of providing high QoS to customers and the migration from under load PM for the less energy consumption in CCSs.

VMC methodology deals with four categories [61] as following.

1. Identifying when a PM can be considered as overloaded, then migrate one or more available VMs present in this PM to other active or reactivated hosts to avoid SLAv.
2. Identifying when a PM can be considered as underloaded, then migrate all the available VMs present in this PM and switch it to sleep mode.
3. Selecting the VMs that need to be migrated from an overloaded PM.
4. Placing the VMs which are selected for migration from both overloaded and underloaded PMs on other active or reactivated PMs.

So, VMC methodology is divided into four subproblems, i.e., four categories of algorithms (each category represents a class of algorithm) as following:

1. PMs overload detection.
2. PMs underload detection.
3. VMs selection.
4. VMs placement. Algorithm for this subproblem can be designed separately to obtain close to an optimal solution which ensures a reduced energy consumption of the active hosts and minimum SLAv.

Finally, VMC methodology comprising two basic approaches: (1) migrating VMs from overloaded PMs to avoid performance reduction and SLAs violation, and (2) migrating VMs from underloaded PMs to optimize utilization of resources and energy consumption with minimizing active PMs. Therefore, inappropriate VMC may lead to performance reduction when an application encounters an increasing demand resulting in an unexpected rise in resource usage. Therefore, the cloud provider has to tradeoff between energy consumption and QoS criteria [36]. So far, various techniques and methods have been proposed for VMC implementation. Many metrics are considered in the presentation of these algorithms and are effective in making decisions; all or many of these metrics can be considered in VMC. So, each method has its objectives. Some of these metrics have direct or indirect effects on each other, but consideration of each of the metrics as the primary purpose can lead to different results [73]. The problem of VMC should be regarded as an optimization problem. There are various resources in PMs, and their utilization should be maximized to reduce electricity consumption. In VMC algorithms, the positioning of VMs on the hosts and the necessary resources for each VM is considered as input, while the new positioning of VMs on the hosts is computed as output [28]. The primary resources in this regard include CPUs, main memories, as well as network and storage devices. If only the processor is considered, for the problem of VMs consolidation, the problem of bin packing can be used. Consideration of several resources converts this problem to a problem of bin packing and adds to the complexity of the problem [59].

The infrastructure of a cloud is composed of different hardware components. The energy consumption of these components of hardware follows different patterns. Nevertheless, in

general, their energy consumption is divided into static and proportional parts. Energy consumption of the machine in idle times is called static energy consumption. As utilization grows, the energy consumed by the machine also increases. This part of the energy consumption, which is proportional to the energy consumption of the machine is called proportional energy consumption. Generally, VMC methods for static energy consumption are considered [49]. The total electric energy consumption of PMs' resources in the CCSs is dependent on CPU, RAM, BW, and DISK. However, there are plenty of surveys indicating that the CPU of PMs consumes more electric energy than the other resources of PMs, and the power consumption and processor utilization have linear relation. A power model for energy consumption defined in formula (1) as follows:

$$P(u) = k \cdot P_{\max} + (1 - k) \cdot P_{\max} \cdot U \tag{1}$$

where $P_{\max}$ is the maximum power consumed when the PM is fully utilized; k is the fraction of power consumed by the idle server (i.e. 70%), and U is the CPU utilization [61]. Minimizing of energy consumption of cloud is one of the objectives is in the VMC. However, a tradeoff between energy consumption and other QoS metrics(software and hardware metrics) in CCSs has been considered. Hardware metrics means the quality of the utilization of resources comprising CPU, RAM, BW, and DISK. Software metrics means the quality of performance, SLAv, load balancing, thermal and heating, cooling, migration overhead, reliability, and the ON–OFF cycle of PMs in CCSs.

3.3 VMs Migration

VMs migration means the migration of VM(s) from a PM as source PM to other PM as target PM along with the best new configuration from the point of view energy consumption, resource utilization, and other QoS in CCSs [16]. Taxonomy of the VMs migration Shown in Fig. 3.

VMs migration methods are used in cloud applications and constitute the basis of VMC. Load balancing, SLAv, VMs affinity, and migration overhead, resource utilization, migration time, numbers of migration are amongst the essential metrics to be considered in VMs migration [74]. Which must be a tradeoff between these criteria, for example, minimizing; migration time, numbers of migration, energy consumption, and SLAv vice versa maximizing, load balancing, and resource utilization. Although, with the migration of a large number of VMs, consequently active PMs and energy consumption decreases, vice versa, the SLAv may be increased. Also, if the migration time of VMs is long, it will cause an interruption in customer service and SLAv. As a result, the cloud provider must pay the penalty for SLAv [26].

3.3.1 VMs Migration Metrics

The essential metrics in VMs migration are comprising load balancing, SLAv, migration overhead, VMs affinity, and migration time defined as following [16, 75].

1. *Load balancing* Services are balanced onto PMs by algorithms, and all resource of PMs utilization comprising CPU, RAM, BW, and DISK, are balanced.
2. *SLAv* Providing services in case of conflict against the agreement between the cloud provider and their customers of the cloud.

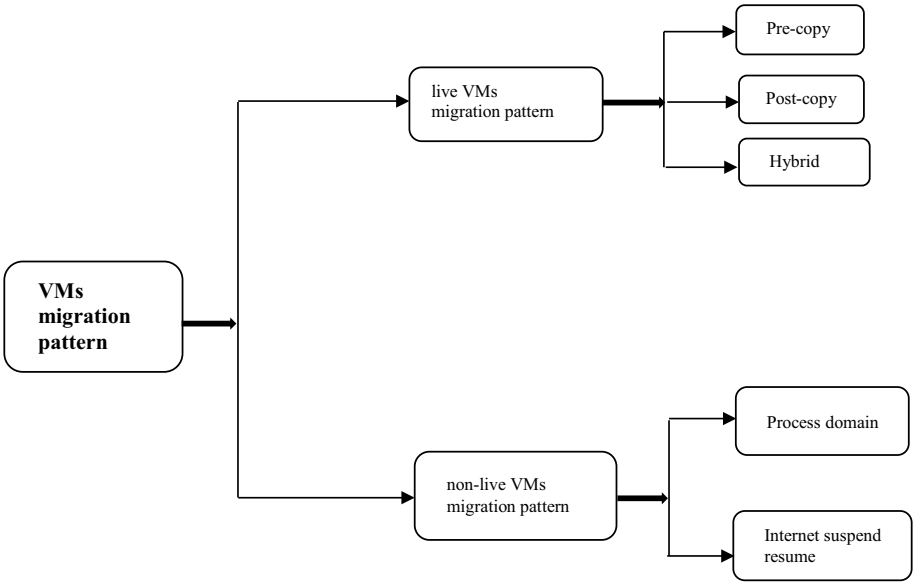

Fig. 3 Taxonomy of the VMs migration

3. *Migration overhead* Some of the criteria harm QoS, such as disproportionate time of migration execution time or the number of migration.
4. *VMs affinity* It refers to the correlation between VMs, as well as the VMs, which depend on each other for executing service.
5. *Migration time* that migration time is the time necessary for transferring a VM from a PM source to a PM destination.

3.3.2 VMs Migration Pattern

VMs migration is comprising: (1) live VMs migration and (2) non-live VMs migration, defined as follows.

1. *Live VMs migration patterns* do not suspend application services during VMs migration and support the running applications during VMs migration time without interruption or minimal interruption. In contrast, in non-live VMs migration methods, application services during VMs migration time suspend [16, 76]. Live VMs migration patterns comprising; pre-copy [49, 77, 78], post-copy [79, 80], and hybrid [31, 81–83]. In the pre-copy method, VM memory pages are iteratively copied until some suitable termination criterion is deduced. In the post-copy method, the captured minimum state (CPU and I/O state) is transferred to the destination PM, where the application serves at the destination PM; then, the rest of the memory pages and resources are transferred. The hybrid method combines the features and benefits of both (pre and post) copy methods to optimize VMs migration mechanism. In comparison to the post-copy method, a hybrid method exploits bounded pre-copy rounds to identify and transfer VM working-set to mitigate the magnitude of the network I/O pages' faults. After completing bounded

pre-copy rounds, the post-copy method transfers VM minimum state to the destination PM to resume VM [81–84].

2. *Non-live VMs migration patterns* do not resume the VM until the VMs are not entirely transferred to the destination PM. In this method, during VM migration, the migrant application services are stopped. Further, non-live VMs migration patterns comprising the Process Domain and Internet Suspend Resume. This method predicates execution migration time and guarantees transfer VM memory pages exactly once during the VMs migration phase [25, 75, 85].

3.4 SLAv

The SLA is the contract between the cloud provider and its customers of the cloud. SLAv in resources level, as the ratio of unallocated resources demanded by applications and the total requested resources. The unallocated resource can be calculated as the difference between the requested resource of all VMs and the actual allocated resources. Further, SLA is agreed upon based on several metrics such as service uptime, delay, response time, throughput, and the service fail rate. Uncontrolled migrations, long time migration, and lack of resources are the main reasons for the SLAv. Cloud providers must be pay penalties to the customer for the SLAv [16].

In addition, to reduce the SLAv; the resources of PMs for users sufficient, Uncontrolled migrations, and migration time must be at least. Live migration has a negative impact on the performance of applications running in a VM during a migration. Therefore uncontrolled migrations must be at least. The average performance degradation, including the downtime, can be estimated as approximately 10% of the CPU utilization [1]. This means that each migration may cause some SLAv; therefore, it is crucial to minimize the number of VM migrations and the length time of live migration. The length time of a live migration depends on the total amount of RAM used by the VM and available NBW. Therefore, migration time is the time necessary for transferring a VM from a PM source to a PM destination. Migration time is calculated as the amount of available RAM divided by available spare network bandwidth (NBW) for PM. Migration time (mt) of a VM_i defined in formula(2) as fallowing [70].

$$mt_{\mathrm{VM}_i} = \mathrm{RAM}_i / \mathrm{NET}_j \tag{2}$$

where RAM_i is the amount of RAM currently utilized by VM_i and NET_j is the spare NBW of PM_j. Which, to the prevention of performance degradation, must be migration time at least. Therefore, migration time and the numbers of migration both are the cost of VMs live migration, there may cause some SLAv and should be least. The essential metrics in SLAv are comprising (1) SLATAP and (2) PDM defined as following [1].

1. *SLA violation Time per Active PM (SLATAP)* The percentage of the time, during which active PMs have experienced the CPU utilization of PM 100%.
2. *Performance Degradation due to Migrations (PDM)* The overall performance degradation by VMs due to migrations.

In addition, migration time and the numbers of migration are two main parameters for calculating to PDM. The reasoning behind the SLATAP is the observation that if a PM serving applications are experiencing 100% utilization, the performance of the applications is bounded by the PM capacity. Therefore, VMs are not being provided

with the required performance level. Finally, the overall SLAv of cloud infrastructure can be captured by combining both of these SLATAP and PDM, defined in formula (3) as follows:

$$\text{SLAv} = \text{SLATAP} \times \text{PDM} \tag{3}$$

4 Papers' Selection Process

This section is the process of selecting the papers for the systematic survey in two steps: (1) Search based on keywords, and (2) selection papers based on the title, abstract, quality of the publisher, and detailed review of papers.

4.1 Search Based on Keywords

The searching context involves VMC and VMs migration, in data centers and CCSs papers. The search process is based on electronic searching methods by considering synonyms of the keywords, the following search string was designed: [("virtual machines migration") OR ("VMs migration") OR ("virtual machines consolidation") OR ("server consolidation") OR ("VMs consolidation")] AND [("cloud") OR ("data-center")], based on online databases, like following famous scientific databases. Table 1 reports these scientific databases used for the automatic search.

This systematic research attempts to respond to the following Technical Questions (TQ) according to the proposed systematic research:

- TQ1: What is the trend of studies published in VMC?
- TQ2: What are the software metrics, hardware metrics, and objectives of VMC?
- TQ3: How (exact, greedy, evolutionary) VMC algorithms are analyzed?
- TQ4: What are the evaluation methods and workload data for VMC?
- TQ5: What are the open issues and challenges in VMC?

101 papers were found using the systematic survey. Figure 4 depicts the distribution of these papers published from 2011 to 2019. The number of published papers in 2019 was the highest. Note that the trend of studies published in VMC has a growing trend.

Table 1 Scientific databases used for the automatic search

Num.	Electronic databases and search engine	Electronic address
1	Google Scholar	scholar.google.com
2	Wiley	onlinelibrary.wiley.com
3	Science Direct	www.sciencedirect.com
4	ACM digital library	dl.acm.org
5	Springer	link.springer.com
6	MDPI	www.mdpi.com
7	IEEE	ieeexplore.ieee.org

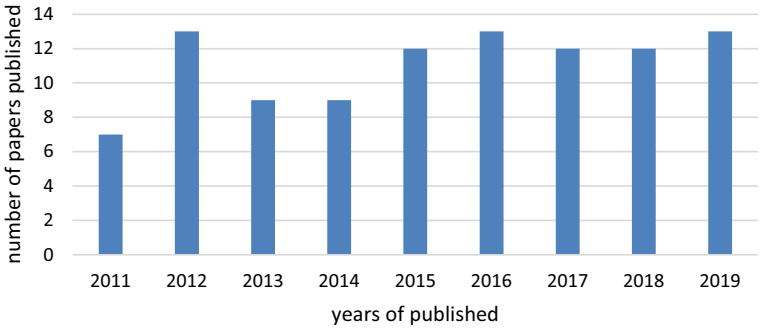

Fig. 4 Distribution of published papers on research scope in terms of years

4.2 Selection Papers Based on the Title, Abstract, Quality of the Publisher, and Detailed Review of Papers

In selection papers, we removed all book reviews, white papers, and non-related subject matters; we excluded 14 conference articles that were not indexed in the scientific organization, 21 not-peer-reviewed papers, and 17 papers not in the Institute of Electrical and Electronics Engineers (IEEE), Springer, Association for Computing Machinery (ACM), Multidisciplinary Digital Publishing Institute (MDPI), Wiley and well- known conferences. Then, we chose only 14 conference articles that were indexed in the IEEE proceedings, Springer, ACM, and articles belonging to well-known conferences, plus 35 peer-reviewed articles indexed in the ISI or Scopus, which were considered for further analysis. Overall, 49 research papers were provided. Figure 5 indicates the percentage of journal and conference papers. Figure 6 illustrates the percentage of the distribution of the journal papers(the distribution of the journal papers according to some famous publishers such as Elsevier, IEEE, Springer, MDPI, ACM, Wiley). In this figure, more articles have been published by Elsevier. Finally, Fig. 7 indicates the selection process of articles.

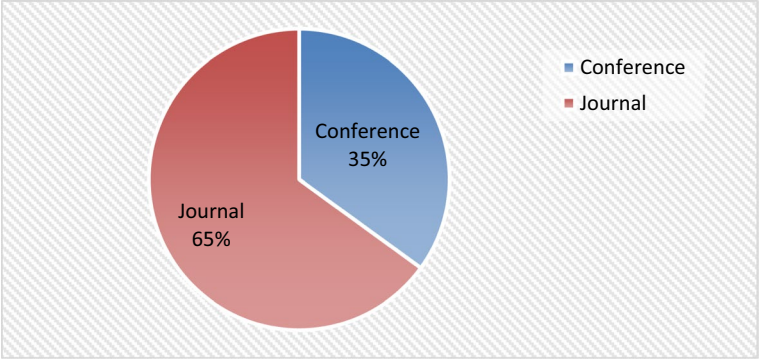

Fig. 5 Percentage of journal and conference articles

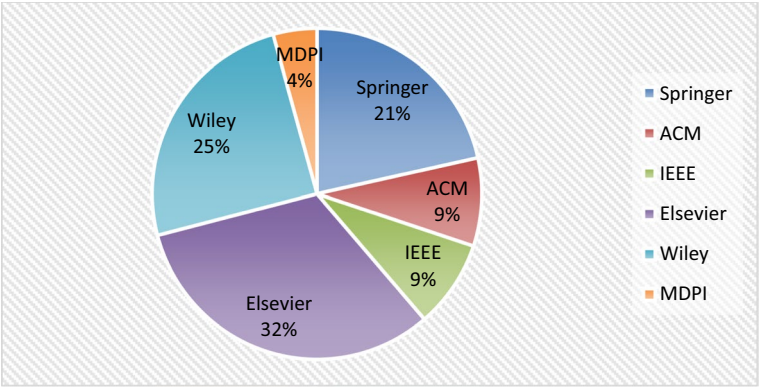

Fig. 6 Percentage of articles by publishers

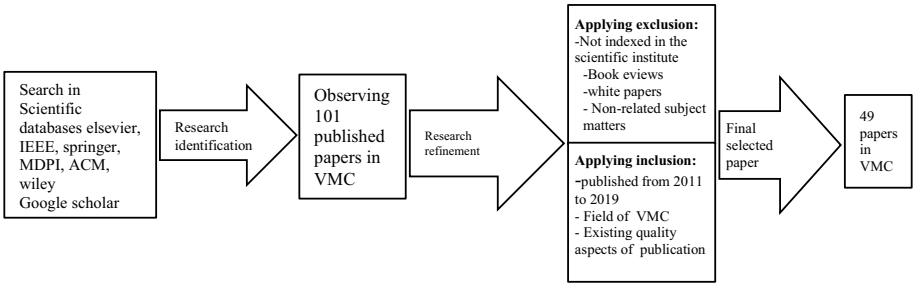

Fig. 7 The selection process of papers

5 A Taxonomy of VMC

Answer to TQ1: This section exhibits the taxonomy and a review of state-of-the-art articles of VMC. A comparison is also made on the existing taxonomy based on the criteria from the technical and relevant literature. This taxonomy comprising; basic concepts of VMC, resource assignment policy, VMC architecture, VMC phases, metric types, main objectives, optimization algorithms, evaluation method in VMC, as portrayed in Fig. 8. This research includes a survey of different VMs migration models, which includes two categories comprising; live VMs migration and non-live VMs migration. Further, VMC methods include two categories of static and dynamic methods based on the policy for dividing the resource assignment. Some several metrics are considered in the taxonomy. Metrics considered to achieve the objectives effectively as decreasing power consumption are presented in Tables 2. Optimization approaches used for solving the VMC in CCSs, comprising; statistical and exact, greedy, and evolutionary methods presented in Tables 3, 4, and 5.

Another essential aspect that should be classified is the data set and the evaluation methods. Data set include synthesis and real data sets. The evaluation methods comprise simulation, implementation, hybrid (simulation and implementation), and formal methods.

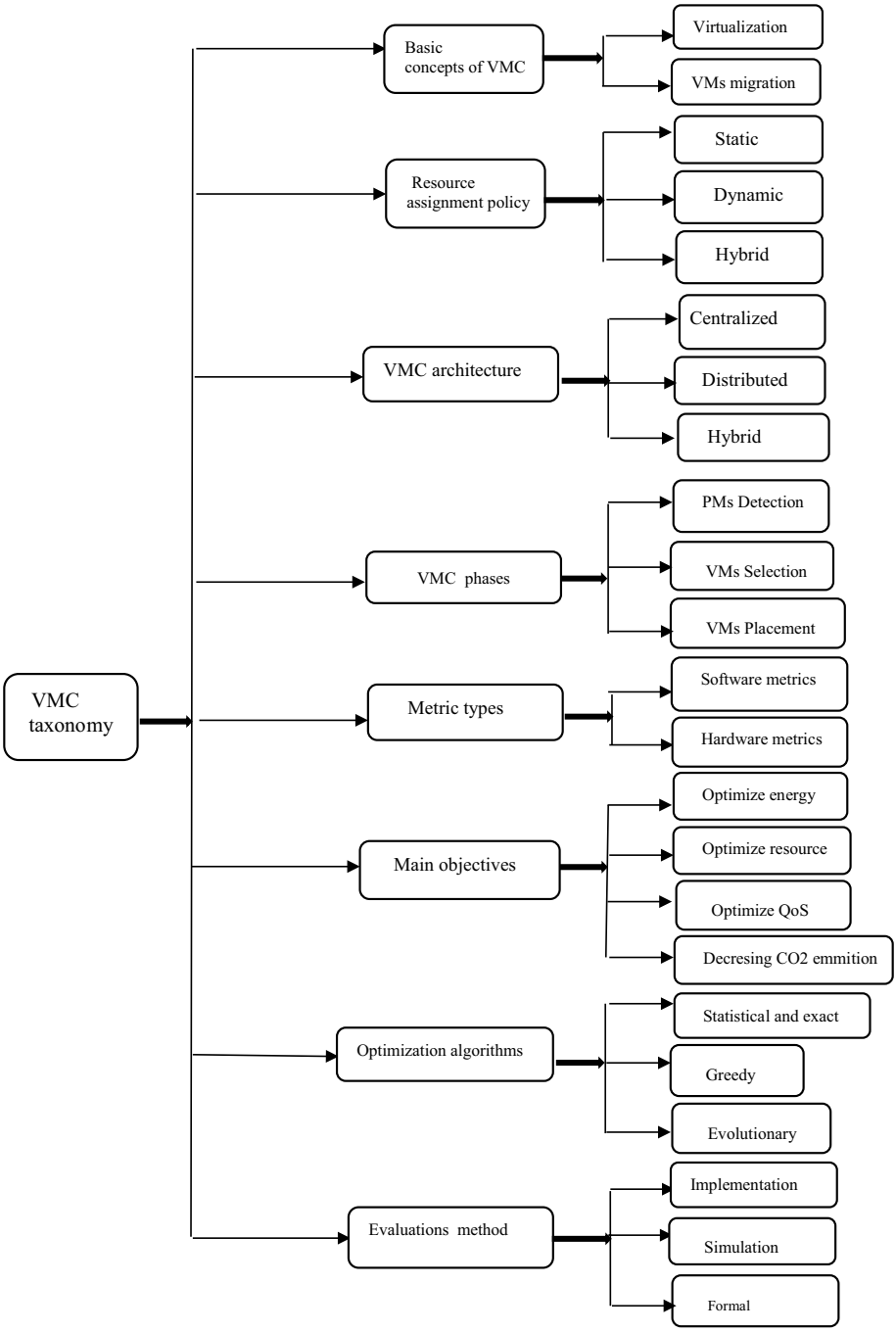

Fig. 8 Taxonomy of the VMC

Table 2 Static/Dynamic VMC

Article	Resource assignment policy		Performance	Cooling	Reliability	ON–OFF cycles	Thermal/ Heating	Data set\Evaluation
	Static	Dynamic						
Pahlavan et al. [5]		✓		✓			✓	Synthesis data/Simulation
Ali Habib, et al. [30]	✓			✓			✓	Synthesis data/Simulation
Garg et al. [36]		✓	✓					Co Mon project/CloudSim toolkit
Mehdi et al. [19]	✓				✓		✓	Real data/Implementation & Simulation
Guenter et al. [20]	✓		✓		✓	✓	✓	Real data/Simulation
Qian, et al. [21]		✓			✓	✓		Synthesis data/Simulation
El-Sayed et al. [22]		✓	✓		✓		✓	Synthesis data/Simulation
Mansour et al. [42]		✓	✓		✓			google cluster/implementation
Hang Zhou et al. [86]		✓	✓					google/cloud sim
Shashikant et al. [37]		✓		✓			✓	SPECpower/CloudSim toolkit
Michael et al. [39]		✓	✓					Synthesis data/Simulation (DC Sim)
Sansottera et al. [87]		✓	✓		✓			SPEC/Simulation(matlab)
Mohan Raj et al. [38]	✓			✓			✓	Co. Mon project/(Planet lab/Cloud Sim toolkit)
Deng et al. [65]		✓	✓		✓		✓	Amazon EC2/CloudSim
Garg et al. [36]	✓		✓					Co. Mon project/(Planet lab, Cloud sim toolkit)
Ferreto et al. [88]	✓				✓			(Google data center, TU-Berlin)/Implementation
Mahfuzur et al. [89]	✓		✓					Google/Simulation (cloud sim)
Kyungmee et al. [68]	✓		✓					Real data/Simulation

Table 3 VMC with statistical and exact methods

Paper	Architecture		Hardware metric				Software metric				
	Centralized	De-centralized	RAM	CPU	NBW	Disk	Migration overhead		Load balancing	VMs affinity	SLAv
							Time	Number			
Zhang et al. [28]	✓		✓	✓		✓					
Huang et al. [48]	✓			✓					✓		✓
Sindlar et al. [61]		✓	✓	✓						✓	✓
Cao et al. [27]	✓				✓			✓			
Ammar Al-Moalmi et al. [63]	✓		✓	✓			✓	✓			
Oshin Sharma et al. [46]		✓		✓					✓	✓	✓
Alain Tchana et al. [49]		✓		✓	✓		✓	✓			
Robayet Nasim et al. [69]	✓			✓							✓

Table 4 VMC with greedy methods

Article	Architecture		Hardware metric				Software metric				
	Centralized	Distributed	RAM	CPU	NBW	Disk	Migration overhead		Load balancing	VMs affinity	SLAv
							Time	Number			
Alicherry et al. [43]		✓			✓			✓			
Fikru Feleke Moges et al. [52]	✓			✓			✓				✓
Kokadia et al. [57]		✓			✓		✓	✓			
Beloglasov et al. [90]	✓			✓				✓			✓
Zhang et al. [58]	✓		✓	✓				✓			
Farahanakian et al. [62]		✓	✓								✓
Changhyeon et al. [31]		✓	✓	✓		✓	✓	✓			
Rashmi Rai et al. [91]	✓			✓					✓	✓	✓
Somnath Mazumdar et al. [34]		✓		✓			✓				
Sedaghat et al. [50]		✓	✓	✓	✓			✓	✓		
Usmani et al. [32]	✓								✓		✓
Terra-Neves et al. [92]		✓	✓	✓							✓

Table 5 VMC with evolutionary methods

Paper	Architectures		Hardware metric				Software metric				
	Centralized	Distributed	RAM	CPU	NBW	Disk	Migration overhead		Load balancing	VMs affinity	SLAv
							Time	Number			
Antonio et al. [66]		✓	✓	✓	✓		✓				
Sonklin et al. [53]		✓		✓				✓			
Ferdaus et al. [54]		✓	✓	✓	✓						✓
Zheng et al. [59]	✓		✓	✓	✓		✓	✓			
Marzolla et al. [60]		✓							✓		
Cao et al. [29]		✓		✓	✓		✓	✓		✓	
Masoumzadeh et al. [24]		✓		✓					✓		✓
Li et al. [25]	✓			✓	✓				✓		
Huixiao et al. [64]		✓	✓	✓					✓	✓	✓
Perla Ravi Theja et al. [41]	✓			✓							✓
Li et al. [55]		✓		✓		✓		✓	✓		✓

Finally, the VMC architecture, involving centralized and distributed architecture. Also, the hybrid architecture (the combination of centralized and distributed architecture) can be used in VMC.

5.1 Resource Assignment Policy in VMC

VMC based on resource assignment policy can be done by two methods comprising; (1) static method and (2) dynamic method, which defined as following.

1. *Static VMC method* VMC method is an efficient approach, which is used by CCSs to improve resource utilization and minimize the energy consumption of the cloud. The VMC process is especially essential when there are unpredictable customer workloads that need to be revisited often. Whenever a change occurs in customer demand, the required VMs can be resized and relocated to other PMs according to the demand. This consolidation process can either be performed in one step by making use of the peak load demands of each customer workload then configure VM capacities accordingly. By using the peak load demand utilization, it guarantees that it avoids overloading of VMs. However, since the workloads have a chance of presenting any variable demand patterns, it may lead to the idleness of VMs. This approach is known as static VMC. In this type of consolidation, VMs are placed in a PMs, and no VM migration takes place, so the VMs remain in the same PM during their entire lifetime [36].
2. *Dynamic VMC method* In dynamic VMC is periodically re-evaluating the workload demand of each VM and performing the appropriate configuration changes. Since it dynamically changes VM capacities depending on the current workload demands, this approach results with better consolidation. This is known as dynamic VMC; in this type of consolidation, VMs are placed into PMs, and if the necessity occurs, the VMs can be migrated to other PMs. In other words, this method continuously re-configures the VMs based on the workload demanded. So, this algorithm has greater flexibility and efficiency versus a static algorithm [27].

The VMC metrics in some papers are listed in Tables 2. The mentioned metrics are included performance, cooling systems, reliability, ON–OFF power cycles, and thermal/heating. Studies' goal is to achieve the decreasing energy consumption, but in these studies, several mentioned metrics have been covered, and some are not covered (those that are left blank). The covered metrics in these tables as advantages and while not covered metrics (those that are left blank) are as disadvantages of these studies. In other words, because these metrics are effective and essential in VMC, consider them are as the advantages vice versa the lack of consideration (those that are left blank) as the disadvantages of these studies. Metrics definition with detail in Sect. 5.2 presented.

5.2 Metrics and Objectives in VMC

Answer to TQ2: VMC metrics and objectives have overlap in many cases. It is assumed that the metrics are independent variables while the objectives are dependent variables; therefore, the objectives can be a result of a combination of two or more metrics. So, optimization of the metrics can also cause the optimization of the objectives. In other words, to implement the VMC algorithms for reducing power consumption as the main

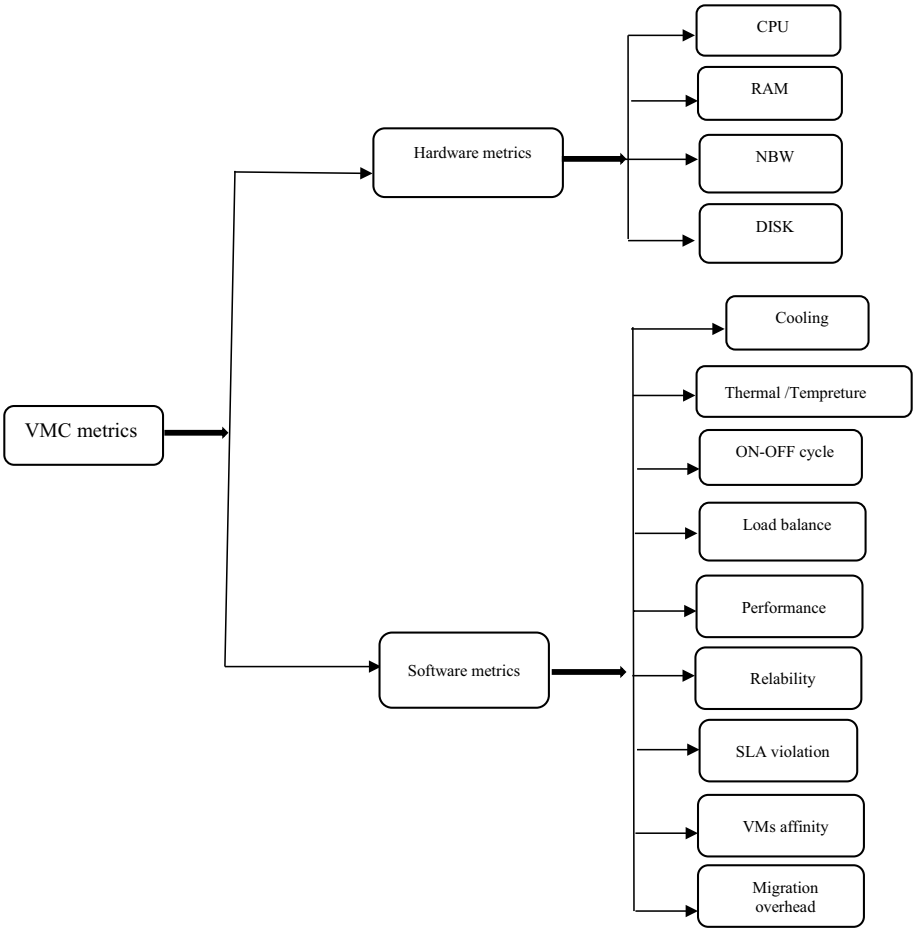

Fig. 9 Taxonomy of the VMC metrics

objective, other software metrics and hardware metrics with it must be considered for a holistic approach. Taxonomy of the VMs metrics Shown in Fig. 9.

Software metrics of VMC comprising performance, thermal, cooling systems, reliability, ON–OFF power cycles, SLAv, VMs affinity, load balancing, and migration overhead. Hardware metrics of VMC comprising as resource utilization of PMs in terms of CPU, RAM, NBW, and HDD. On the other hand, minimizing: migration execution time, active host, CO_2 emission, Thermal, ON–OFF cycles, and maximizing: load balancing, performance, resource utilization of PMs in terms of CPU, RAM, NBW, and HDD are amongst metrics to VMC algorithms. Figure 8 shows a taxonomy of the VMC metrics. Figure 10 and 11 respectively illustrate the percentage related to each software metrics and hardware metrics for VMC algorithm in the studied papers.

The essential software metrics and hardware metrics, which, along with energy reduction as the main objective of VMC, must be considered. It defined as following:

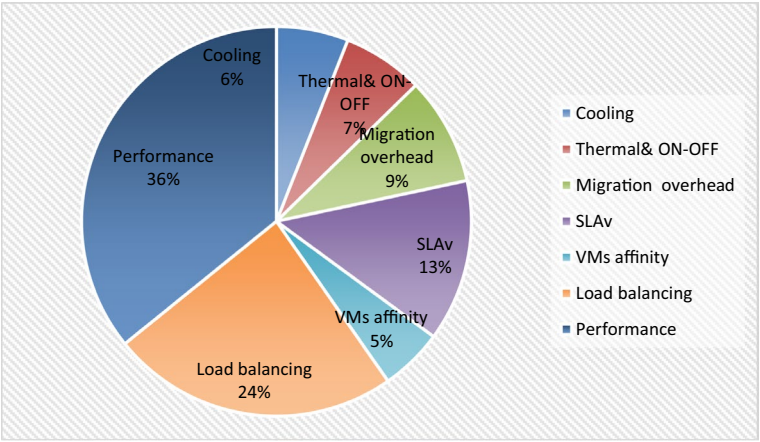

Fig. 10 Percentage of each software metrics of VMs consolidation considered in the studied papers

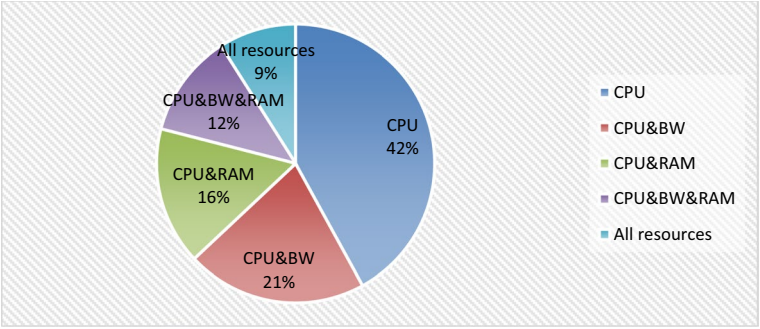

Fig. 11 Percentage of each hardware metrics of VMs consolidation considered in the studied papers

1. *Thermal/Heating and ON–OFF power cycles* High temperatures can lead to several problems such as diminished system reliability and availability, shortened hardware lifespan, a continuous increase in operational and hardware costs, as well as SLAv and their penalty. To keep the system components within their safe operating temperature and prevent failures and crashes, the emitted heat must be dissipated. These PM' ON–OFF power cycles are known as the essential factor of the average residual time of a storage failure [21, 39, 40].
2. *Cooling* The cooling system is one of the most widely used devices in CCSs. Minimizing their usage can optimally reduce energy consumption. Recent research suggests that about 50% of the energy consumption of CCSs is related to its cooling equipment. Hence, by optimizing the cloud data center's performance, the energy consumption of the cooling system can be reduced, ultimately resulting in diminished CCSs energy consumption [5, 38].
3. *Performance* One of the metrics contributing to the diminished efficiency of a PM is inter-VM performance degradation of VMs located on a PM. Virtualization does not provide a guarantee for separating the efficiency among different VMs that work on a

server. Therefore, while VMs store a series of resources, their simultaneous execution on PMs might cause competition and performance interference [34].

4. *Reliability* VMC can harm the longevity of the cloud PMs. VMC methods try to mount the data centers' VMs on a fewer number of PMs and turn off the idle PMs, which is mainly because the capacity of these inactive PMs may be required again in the future. Therefore, more PMs are needed for restarting and placing some VMs on them. This turn ON–OFF cycles can harm hardware lifetime [40]. Meanwhile, VMC dramatically raises resource efficiency, which will increase the PM temperature. This, in turn, can reduce the reliability and longevity of PMs. Reduction of the lifespan will result in hardware errors, which will reduce the reliability of services [21].
5. *Hardware utilization* Resources utilization of PM are comprising CPU, RAM, NBW, and HDD as hardware metrics of VMC method [32]. The authors [36], only considered the CPU for the VMC algorithm. In [49], the authors considered both CPU and RAM as metrics. Finally, the authors [65], considered all CPU, RAM, NBW, and HDD resources. In [23, 43, 66], the authors considered the NBW for the VMC algorithm. The amount of traffic flow on the network and the extent of inter-VMs communication are the most influential factors for the QoS and services performance. Also, in batch processing, the prolongation of the communication time between two elements can increase the runtime of tasks, thereby augmenting the energy consumption.

5.3 Architectures in VMC

VMC architecture is consists of centralized and distributed. Centralized VMC Architecture is prone to Single Failure Point (SFP). It is unreliable, while the distributed architecture approach makes the cloud more scalable and extensible as there is no risk of an SFP [74]. A combination of centralized and distributed architectures as hybrid architecture is also practical. These architectures are used to Optimization methods in VMC, but distributed architecture more used.

5.4 Optimization Methods in VMC

Answer to TQ3: Optimization methods for VMC comprising three categories comprising (1) statistical/exact methods, (2) greedy methods, and 3)evolutionary methods [9, 56]. Figure 12 illustrates the percentage of each optimization method of VMC considered in the studied papers. As well, the taxonomy of the Optimization methods in VMC Shown in Fig. 13.

Exact methods guarantee an optimal solution for problem-solving. This approach is suitable for solving the problem that belongs to the class P or the NP-hard problems with (n) size [28]. Generally, greedy methods and evolutionary methods are used for problem-solving. Considered metrics of optimization methods in VMC, which, mentioned in the Tables 3, 4 and 5 include; the resources utilization of PMs comprising RAM, CPU, NBW, and HDD as hardware metrics, and metrics are comprising; migration overhead (time of migration and the number of migration), SLAv, VMs affinity, and load balancing as software metrics presented. Main objectives, in all of the studies, this section is the reducing power consumption with VMC algorithms, but in these studies, several mentioned metrics have been covered as an advantage, and some are not covered (those that are left blank) as the disadvantage of the research approach. In other words, because these metrics are effective and essential in VMC, considering them are as the

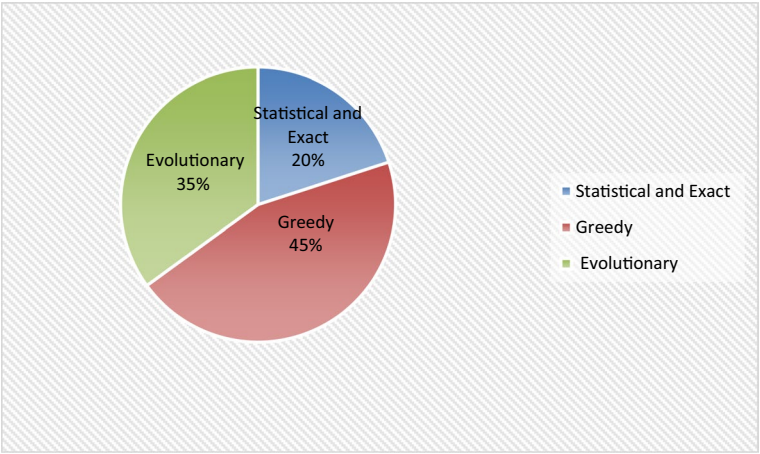

Fig. 12 Percentage of each optimization method of VMC considered in the studied papers

advantages vice versa, the lack of consideration (those that are left blank) as the disadvantages of these studies. These three category algorithms defined as following.

1. *Statistical and exact method* At that, mathematical modeling and then optimal algorithms are designed for the solution. These algorithms can solve and obtain the exact answer for small-sized. Linear programming [48], dynamic programming [15], and Constraint Satisfaction Problems (CSP) [49] are among the most widely used methods for problem-solving in optimal VMC of CCSs in mathematical or statistical methods. Table 3 presents several studies that have used exact methods, which covered metrics as an advantage and not covered metrics (those that are left blank) as a disadvantage of the research approach.
2. *The greedy method* is an NP-hard optimization and multi-dimensional bin packing problem and has demonstrated high performance for solving NP-hard problems [52]. Greedy methods are problem-dependent methods which do guarantee to find the optimal solution; instead of finding a global optimum, they might find optimal local results. Also, these methods try to find a near-optimal solution within optimal time. In this method, greedy algorithms such as FFD [50], BFD [34], and MBFD [52] have been utilized. Table 4 presents several studies that have used greedy methods, which covered metrics as an advantage and not covered metrics (those that are left blank) as a disadvantage of the research approach.
3. *The evolutionary method* is an approximate optimization approach used for solving optimization problems in VMC. Unlike greedy methods, these methods guarantee to find a global optimal point. Indeed, evolutionary methods are practical approaches for finding optimal or near-optimal solutions for problem-solving and generally need more time for problem-solving in comparison with greedy methods [53]. Amongst the evolutionary methods, the GA [41, 53], ACO [54], and PSO [55] can be observed. Table 5 presents several studies that have used evolutionary methods, which covered metrics as an advantage and not covered metrics (those that are left blank) as a disadvantage of the research approach.

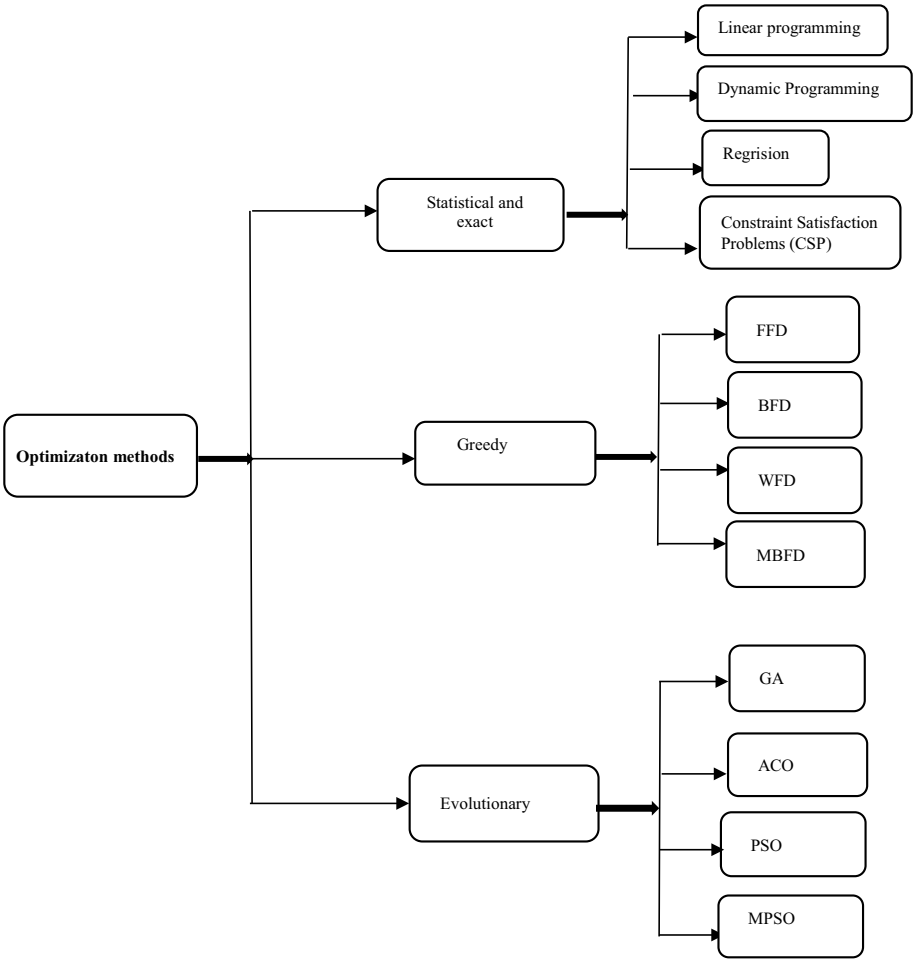

Fig. 13 Taxonomy of the Optimization methods in VMC

5.5 Evaluation Methods and Data Set of VMC

Answer to TQ4: a dataset for VMC is comprising; synthetic data set, real data set, and hybrid data set. Indeed, a hybrid data set is a combination of synthetic and real data sets. These are instant data for evaluating the proposed algorithms.VMC evaluation is one of the main challenges in the cloud. The VMC evaluation has been classified comprising: implementation, simulation, hybrid (both implementation and simulation), and formal methods. Figure 14 illustrates the percentage of evaluating methods used in studied papers.

The complexity of cloud processes and user interactions in cloud environments is progressively increasing with the advancement of technology. Therefore, simulation and implementation methods have not been suitable and effective approaches for evaluating complex CCSs. Accordingly, because of the high complexity of the VMC problem, as the operations in a cloud system are real-time, the formal method is very suitable for evaluation in the cloud, such as VMC in the cloud. Thus, the formal methods can be used to

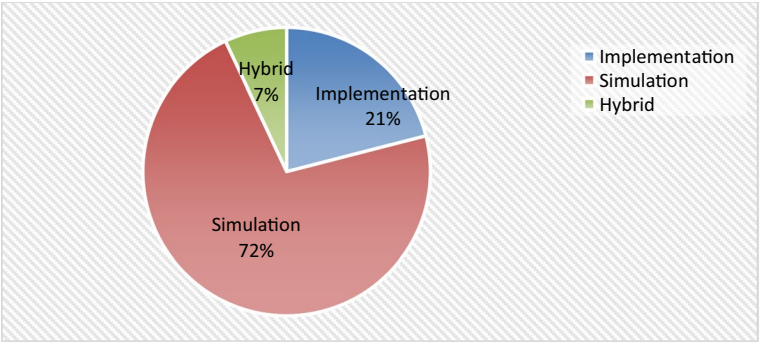

Fig. 14 Percentage of evaluating methods used in papers

precisely evaluate the cloud, and particularly evaluating the functional properties of CCSs. Formal methods have been categorized into three disciplines; process algebra [93, 94], model checking [95], and theorem proving [96, 97], which have been used in the assessment of many specific systems cloud.

Several studies in cloud comprising security [98–100], big data [101, 102], task scheduling [103], migration overhead [104], mobile computing [105], service composition [106] have been conducted for evaluation of open research issues and challenges via formal methods. Table 6 reports the data set and simulation tools used in the reviewed papers. In most studied papers, the planet lab was used as the data set platform [107]. As well, the taxonomy of the formal methods in VMC Shown in Fig. 15.

6 Challenges and Technical Issues in VMC

Answer to TQ5: the VMC comprising; Problem definition and optimal algorithms for problem-solving.

Table 6 Workload dataset and evaluating tools used in studied researches

Data set(Synthesis data, Real data)	Simulation tools
Amazon	Open stack
Amazon EC2	Cloud SIM
Spree-commerce	Peer SIM
TB-berlin	Green cloud
Bit brains	EMU SIM
Specjbb-TPC	MDS SIM
Specjms-TPC	Cloud analyst
Lamp benchmark	Open nebula
Net-prof-UDP	Snoose
Post mask	Cloud SIM toolkits
Google cluster	Network cloud SIM
Spectweb2005bank	
Planet lab/Co Mon project	

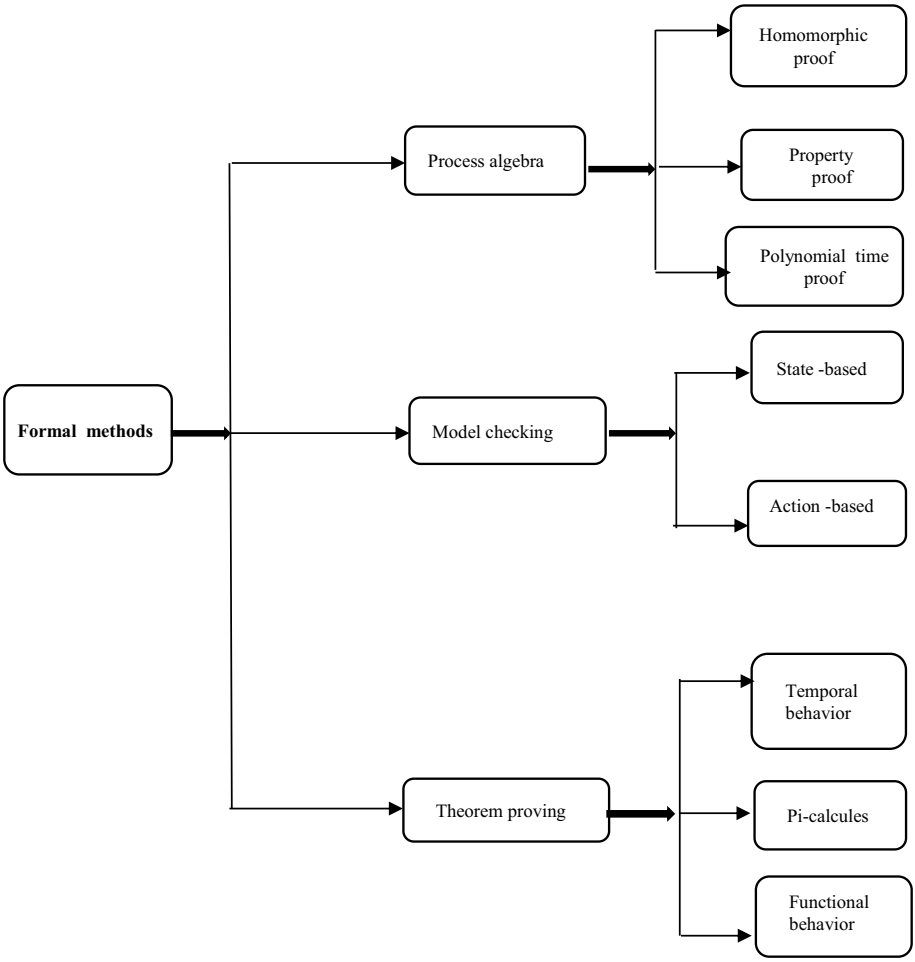

Fig. 15 Taxonomy of the formal methods in VMC

Problem definition in VMC Most of the research in VMC has focused mainly on a few constraints, such as the utilization of CPU infrastructure resources in CCSs, which are not comprehensive approaches to capture all infrastructure resources of PMs and types of equipment in the cloud. Implementation of VMs consolidation while considering all resources of PMs in CCSs needs more efficient and optimal solutions.

Storages must be considered in the VMC of CCSs. In some CCSs, centralized storage devices have been used, such as storage area network (SAN), while in some CCSs, distributed storage devices have been employed, including storage on PMs. Two methods reflect different behaviors in the context of power consumption for running the VMC. As well, BW and distance between the PMs, the homogeneity/non-homogeneity of the VMs and PMs, the Inter-correlation between the VMs and the correlation between the VMs and PMs in VMC are considerable. As well, the coefficient between the VMs and resources of PM in VMC is considerable.

Although all VMC metrics and objectives are essential, some of them are in contrast to each other. For example, maximizing resource utilization is in contrast to minimizing thermal rise through cooling devices. Thermal and ON–OFF power cycles are in contrast to reliability, service availability, hardware cost, and hardware longevity. A load balancing degree is in contrast to the resource utilization rate and power consumption. Therefore, VMC approaches should combine multiple objectives for satisfying the objective final. Combine multiple objectives, it is possible to cause the increasing complexity of problem-solving.

In addition, big data, mobile computing, application performance, storage type, BW, resource utilization rate, SLAv evaluation, migration time, the number of migration, task scheduling, service composition, and security are open research issues and challenges of VMC in CCSs.

Problem-solving in VMC With the growth of online services, the expansion and complexity of CCSs inevitably exert more pressure on VMC algorithms to provide scalability in CCSs. Therefore, there is an increasing need for techniques that are distributed, dynamic, hierarchical, exact, and quick in VMC. Some significant research challenges about VMC are: How to effectively combine different optimal algorithms? Can hierarchal techniques, greedy algorithms, evolutionary algorithms, and hybrid algorithms help in this regard? What are the functional criteria and non-functional criteria in VMC? How the functional criteria and non-functional criteria in VMC can be evaluated? What is the optimal time for running the algorithms? How can parallel algorithms be used for speeding up the runtime of algorithms? How can distribute, dynamic and scalable algorithms be utilized for the complexity of problem-solving in VMC? How can the formal method be used to evaluate VMC?

7 Conclusion and Future work

This research is presented with a systematic review of the VMC methods in the cloud. A systematic technique was used for optimal searches, and 49 relevant papers in VMC were selected. Also, the taxonomy for VMC, including; VMC phases, VMC methods in terms of decision time, migration patterns, criteria and objectives, optimization algorithms, architectures, data set, and evaluation methods for VMC, were presented. Besides, the software metrics in VMC include thermal and ON-OFF power cycles, cooling systems, performance, reliability, resource utilization, listed according to Tables 2 in some of the previous works by two methods of static VMC and dynamic VMC were stated.

VMC phases comprising: (1) PMs detection, (2) VMs selection, and (3) VMs placement and the necessary algorithms for all three phases are explained with detail. The necessary algorithms and policies are comprising three disciplines exact method, greedy method, and evolutionary method.

VMC is considered an essential technique for resolving the tradeoff between energy consumption and QoS metrics in the cloud. VMC has various metrics, objectives, and different algorithms. The variability of the metrics and algorithms can result in a variety of objectives. VMC is one of the open issues in CCSs, such as its evaluation. Evaluation of proposed VMC techniques in CCSs has been mainly done with simulation and implementation or hybrid (simulation and implementation together). Rarely, the formal method has been used in the cloud, while, because of the high complexity and real-time operations in the cloud system, the formal method is very suitable for the evaluation of cloud, such as

VMC in the cloud system. Therefore, research on formal methods for modeling, evaluating, and verifying the approaches of VMC and other cloud terms, especially functional properties of algorithms in VMC.

Note that cloud providers may need to manage trade-off of energy consumption, QoS, and SLAv through the penalty of SLAv and to develop a comprehensive solution by combining several allocation techniques with different objectives for VMC algorithm in the cloud. In addition, the determination of the threshold for resource utilization of the PMs is an essential solution for the VMC algorithm, then greedy and evolutionary algorithms used for VMs placement. Therefore, research for approaches of determining upper/lower threshold points for utilization of the PMs for VMC, greedy, and evolutionary algorithms used for VMs placement in CCSs was suggested. They would mainly involve the dynamic determination of the upper/lower threshold for all resources of PMs comprising CPU, RAM, NBW, HDD, etc. Finally, the lack of resources in PMs and uncontrolled migration of VMs cause a SLAv in the cloud, which is an essential issue to consider. Therefore, cloud providers have to tradeoff between energy consumption and QoS criteria, especially SLAv.

References

1. Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud computings. *Concurrency and Computation: Practice and Experience, 24*(13), 1397–1420.
2. Ashraf, A., Porres, I., Naeen, H. M., Zeinali, E., & Haghighat, A. T. (2018). A stochastic process-based server consolidation approach for dynamic workloads in cloud data centers. *The Journal of Supercomputing, 76*(3), 1903–1930.
3. Qiu, Y., Jiang, C., Wang, Y., Ou, D., Li, Y., & Wan, J. (2019). Energy aware virtual machine scheduling in data centers. In *Energies, MDPI.*
4. Xie, L., Chen, S., Shen, W., & Miao, H. (2018). A novel self-adaptive vm consolidation strategy using dynamic multi-thresholds in IaaS Clouds. In *Future Internet, MDPI* (pp. 1–18).
5. Pahlavan, A., Momtazpour, M., & Goudarzi, M. (2014). Power reduction in HPC data centers: A joint server placement and chassis consolidation approach. *The Journal of Supercomputing, 70,* 845–879.
6. Roytman, A., Kansal, A., Govindan, S., Liu, J., & Nath, S. (2013). PACMan: Performance-aware virtual machine consolidation. In *10th international conference on autonomic computing (ICAC2013)* (pp. 83–94).
7. Ullah, A., Li, J., Shen, Y., & Hussain, A. (2018). A control theoretical view of cloud elasticity: Taxonomy, survey and challenges. *Cluster Computing, 21,* 1735–1764.
8. Witanto, J. N., Lim, H., & Atiquzzaman, M. (2018). Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management. In *Future generation computer systems* (pp. 1–20). Elsevier, New York.
9. Casalicchio, E., Lundberg, L., & Shirinbab, S. (2017). Energy-aware auto-scaling algorithms for Cassandra virtual datacenters. *Cluster Computing, 20,* 2065–2082.
10. Md Khan, A., Paplinski, A. P., Khan, A. M., Murshed, M., & Buyya, R. (2018). Exploiting user provided information in dynamic consolidation of virtual machines to minimize energy consumption of cloud data centers. In *Third international conference on fog and mobile edge computing (FMEC)*, IEEE.
11. World Energy Outlook. (2013). Fact Sheet. http://goo.gl/Fxl.639.
12. Zhu, R., Sun, Z. & Hu, J. (2012) Special section: Green computing. In *Future generation computer systems* (Vol. 28, pp. 368–370). Elsevier, New York.
13. Asad, Z., & Chaudhry, M. A. R. (2016). A two-way street: green big data processing for a greener smart grid. *IEEE Systems Journal, 99,* 1–11.
14. Shehabi, A., Josephine, S. S., Sartor, D. A., Brown, R., Herrlin, E. M., Koomey, J. G., Masanet, E. R., Horner, N., Azevede, I. L., & Limtner, W. (2016). United States data center energy usage report. Lawrence Berkeley National Laboratory, Berkeley, CA.
15. Kumar, S., Deepak, M., & Bibhudatta, P. (2018). Energy-Efficient VM-Placement in Cloud Data Center. *Sustainable Computing: Informatics and Systems, 20,* 48–55.

16. Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual Machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications, 52,* 11–25.
17. Agar, M. (2013). Developers, engaging the missing link in IT resource efficiency. The Green Grid.
18. Barroso, L. A., Clidaras, J., & Hölzle, U. (2013). *The Data center as a Computer An Introduction to the Design of Ware house- Scale Machines* (2nd ed.). San Rafael: Morgan and Claypool Publishers.
19. Mayahi, M. R., Rezazad, M., & Azad, H. S. (2018). Temperature-aware power consumption modeling in Hyper scale cloud data centers. *Future Generation Computer Systems, 94,* 130–139.
20. Guenter, B., Iain, N., & Williams, C. (2011). Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning. In *INFOCOM, 2011, Proceedings IEEE* (pp. 1332–1340).
21. Qian, H., & Medhi, F. (2011). Server operational cost optimization for cloud computing service providers over a time horizon. In *Proceedings of the 11th USENIX conference in Hot topics in management of Internet, cloud, and enterprise networks and services*.
22. El-Sayed, N., Stefanovici, I. A., Amvrosiadis, G., Hwang, A. A., & Schroeder, B. (2012). Temperature management in data centers: why some (might) like it hot. *ACM SIGMETRICS Performance Evaluation Review, 40,* 163–174.
23. Bodik, P., Menache, I., Chowdhury, M., Mani, P., Maltz, D. A., & Stoica, I. (2012). Surviving failures in bandwidth-constrained datacenters. In *Proceedings of the ACM SIGCOMM 2012 conference in applications, technologies, architectures, and protocols for computer communication* (pp. 431–442).
24. Masoumzadeh, S. S., & Hlavacs, H. (2015). A cooperative multi-agent learning approach to manage physical host nodes for dynamic consolidation of virtual machines. In *2015 IEEE fourth symposium in network cloud computing and applications (NCCA), IEEE*.
25. Li, Z., Yan, C., Xinrong, Yu., & Ning, Yu. (2017). Bayesian network-based Virtual Machines consolidation method. *Future Generation Computer Systems, 69*(2017), 75–87.
26. Xu, H., Liu, Y., Wei, W., & Xue, Y. (2019). Migration cost and energy-aware virtual machine consolidation under cloud environments considering remaining runtime. *International Journal of Parallel Programming, 47*(3), 481–501.
27. Cao, J., Ma, Z., Xie, J., Zhu, X., Dong, F., & Liu, B. (2017). Towards tenant demand-aware bandwidth allocation strategy in cloud datacenter. *Future Generation Computer Systems, 105,* 904–915.
28. Zhang, X., Qiu, L., Qian, Q., & Li, Y. (2015). Virtual machines consolidation and placement based in constraint satisfaction in the clouds. *Journal of Computational Information Systems, 10*(7), 5251–5258.
29. Cao, B., Gao, X., Chen, G., & Jin, Y. (2014). NICE: Network-Aware VM consolidation scheme for enter conservation in data centers. In *Proceedings of the 20th IEEE international conference in parallel a distributed system (ICADS), Hsinchu, Taiwan, IEEE*.
30. Khalaja, A. H., & Halgamuge, S. K. (2017). A Review on efficient thermal management of air- and liquid-cooled data centers from chip to the cooling system. *Applied Energy, 205,* 1165–1188.
31. Kim, C., & Jeon, C. (2015). A parallel migration scheme for fast virtual machine relocation on a cloud cluster. *Journal of Supercomputing, 71,* 4623–4645.
32. Usmani, Z., & Singh, S. (2016). A survey of virtual machine placement techniques in a cloud data center. In *International conference on information security & privacy (ICISP), IEEE*.
33. Aryania, A., Aghdasi, H. S., & Khanli, L. M. (2018). Energy-aware virtual machine consolidation algorithm based on ant colony system. *Grid Computing* (pp. 477–491). New York: Springer.
34. Mazumdar, S., & Pranzo, M. (2017). Power efficient server consolidation for Cloud data center. *Future Generation Computer Systems, 70,* 4–16.
35. Halder, K., Bellur, U., & Kulkarni, P. (2012). Risk-aware provisioning and resource aggregation based consolidation of virtual machines. In *5th IEEE international conference in cloud computing (CLOUD)* (pp. 598–605).
36. Garg, S. K., Toosi, A. N., Gopalaiyengar, S. K., & Buyya, R. (2014). SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. *Journal of Network and Computer Applications, 45,* 108–120.
37. Ilager, S., Ramamohanarao, K., & Buyy, R. (2019). ETAS: Energy and thermal aware dynamic virtual machine consolidation in cloud data center with proactive hot spot mitigation. *Concurrency and Computation: Practice and Experience, 31*(17), e5221.
38. Kumar, M. R. K., & Raghunathan, S. (2015). Heterogeneity and thermal-aware adaptive heuristics for energy efficient consolidation of virtual machines in Infrastructure clouds. *Journal of Computer and Systems Science, 82*(2), 191–212.
39. Tighe, M., & Bauer, M. (2017). Topology and application aware dynamic VM management in the cloud. *Journal of Grid Computing, 15*(2), 273–294.

40. Xiao, X., Xie, G., Cheng, X., & Fan, C. (2017). Maximizing reliability of energy constrained parallel applications on heterogeneous distributed systems. *Journal of Computational Science, 26,* 344–353.
41. Theja, P. R., & Khadar Babu, S. K. (2016). Evolutionary computing based on QoS oriented energy efficient VM consolidation scheme for large scale cloud data centers. *Cybernetics And Information Technologies, 16*(2), 97–112.
42. Khelghatdoust, M., Gramoli, V., Sun, D. (2016). GLAP: Distributed dynamic workload consolidation through gossip-based learning. In *2016 IEEE international conference on cluster computing, IEEE.*
43. Alicherry, M., & Lakshman, T. (2012). Network aware resource allocation in distributed clouds. In *Infocom 2012 Proceedings IEEE* (pp. 963–971). IEEE.
44. Malekloo, M.-H., Kara, N., & El Barachi, M. (2018). An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments. *Sustainable Computing, 17,* 9–24.
45. Halder, K., Bellur, U., Kulkarni, P. (2012). Risk aware Provisioning and resource aggregation based consolidation of virtual machines. In *IEEE 5th international conference in cloud computing (CLOUD)* (pp. 598–605).
46. Sharma O, Saini H (2016) VM consolidation for cloud data center using median based threshold approach. In *Computer Science* (pp. 27–33).
47. Pahlavan, A., Momtazpour, M., & Goudarzi, M. (2012). Data center power reduction by heuristic variation-aware server placement and chassis consolidation. In *IEEE 16th CSI international symposium in computer architecture and digital systems (CADS)* (pp. 150–155).
48. Huang, Z., Tsang, D. H. K. (2012). SLA guaranteed virtual machine consolidation for computing clouds. In *Proceeding of the 2012 IEEE international conference on communications (ICC), Ottawa, Ontario, Canada, IEEE.*
49. Tchana, A., De Palma, N., & Safieddine, I. (2016). Software consolidation as an efficient energy and cost saving solution. *Future Generation Computer Systems* (pp. 1–12). New York: Elsevier.
50. Sedaghat, M., Hernandez-Rodriguez, F., & Elmroth, E. (2016). Decentralized cloud datacenter reconsolidation through emergent and Topology-aware behavior. *Future Generation Computer Systems, 56,* 51–63.
51. Wang, J. V., & Ganganath, N. (2018). Bio-inspired heuristics for VM consolidation in cloud data, centers. *IEEE Systems Journal, 14*(1), 152–163.
52. Moges, F. F., & Abebe, S. L. (2019). Energy-aware VM placement algorithms for the Open Stack Neat consolidation framework. *Journal of Cloud Computing, 8*(1), 2.
53. Sonklin, C., Tang, M., & Tian, Y. C. (2017). New decrease-and-conquer strategies for the dynamic genetic algorithm for server consolidation. In *Neural information processing, ICONIP* (Vol. 10637). Springer, New York.
54. Ferdaus, M. H., Murshed, M., Calheiros, R. N., Buyya, R. (2014). Virtual machine consolidation in cloud data centers using ACO metaheuristic. In *Proceedings of the 20th international conference in parallel proceedings Euro- Porto Portugal*. Springer, New York.
55. Li, H., & Zhu, G. (2015). *Energy-efficient migration and consolidation algorithm, of virtual machines in data centers for cloud computing*. Wien: Springer.
56. Mann, Z. Á. (2015). Allocation of virtual machines in cloud data centers survey of problem models and optimization algorithms. *ACM Computing Surveys, 48*(1), 1–34.
57. Dharmesh, K., Korpi, N., Varma, V. (2013). Network-aware virtual machine consolidation for large data centers. In *Proceedings of the 3rd international workshop on network-aware data management (NDM '13) Denver, CO 2013*.
58. Zhang, X., Tingming, W., Chen, M., Wei, T., Zhou, J., Shiyan, H., et al. (2019). Energy-aware virtual machine allocation for cloud with resource reservation. *The Journal of Systems and Software, 147*(2019), 147–161.
59. Zheng, Q., Li, R., Shah, N., Zhang, J., Tian, F., Chao, K. M., Li, J. (2015). Virtual Machine consolidation placement based on multi-objective biogeography based on optimization. In *Future generation computer system* (pp. 1–28). Elsevier.
60. Marzolla, M., Babaoglu, O., & Panzieri (2011). Server Consolidation in cloud through gossiping. In *Proceeding of the IEEE international symposium in a world of wireless, mobile and multimedia networks, Lucca, Italy, IEEE.*
61. Beloglazov, A., Abawajyb, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data center for Cloud computing. *Future Generation Computer Systems, 28*(5), 755–768.
62. Farahanakian, F., Pahkkala, T., Liljeberg, P., Plosia, J., Tenhunen, H. (2015). Utilization production aware VM consolidation approach for green cloud computing. In *Proceedings of the IEEE 8rd international conference in cloud computing* (New York, IEEE).

63. Al-Moalmi, A., Luo, J., Salah, A., & Li, K. (2019). Optimal virtual machine placement based on grey wolf optimization. In *Electronics, MDPI.*
64. Xiao, H., Zhigang, H., & Li, K. (2019). Multi-objective VM consolidation based on thresholds and ant colony system in cloud computing. *IEEE Transaction Cloud Computing, 7,* 53441–53453.
65. Deng, W., Liu, F., Jin, H., Liao, X., & Liu, H. (2014). Reliability-aware server consolidation for balancing energy-lifetime tradeoff in virtualized cloud datacenters. *International Journal of Communication Systems, Wily, 27,* 623–642.
66. Marotta, A., Avallone, S., & Kassler, A. (2017). *A joint power efficient server and network consolidation approach for virtualized data centers* (pp. 65–80). New York: Elsevier.
67. Gu, L., Zeng, D., & Guo, S. (2015). Joint optimization of VM placement and request distribution for electricity cost cut in geo-distributed data centers. In *2015 international conference on computing, networking and communications, internet services and applications symposium, IEEE.*
68. Chang, K., Park, S., Kong, H., & Kim, W. (2018). Optimizing energy consumption for a performance-aware cloud data center in the public sector. *Sustainable Computing: Informatics and Systems, 20,* 34–45.
69. Nasim, R., Zola, E., & Kassler, A. J. (2018). Robust optimization for energy-efficient virtual machine consolidation in modern datacenters. *Cluster Computing, 21*(3), 1681–1709.
70. Beloglazov, A., & Buyya, R. (2013). Managing overloaded hosts for dynamic consolidation of virtual machines in cloud computing under quality of service constraints. *IEEE Transactions on Parallel Distributed systems, 24*(7), 1366–1379.
71. Papadimitriou, G., Chatzidimitriou, A., & Gizopoulos, D. (2019). Adaptive voltage/frequency scaling and core allocation for balanced energy and performance on multicore CPUs. In *2019 IEEE international symposium on high performance computer architecture (HPCA)* (pp. 133–146). IEEE.
72. Terra-Neves, M., Lynce, I., & Manquinho, V. (2019). Virtual machine consolidation using constraint-based multi-objective optimization. *Journal of Heuristics, 25*(3), 339–375.
73. Khan, A. A., Zakarya, M., & Khan, R. (2018). Energy-aware dynamic resource management in elastic cloud datacenters. *Simulation modelling practice and theory* (pp. 82–99). New York: Elsevier.
74. Ahmad, R. W., et al. (2015). Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. *J. Supercomput., 71*(7), 2473–2515.
75. Yao, L., Wu, G., Ren, J., Zhu, Y., & Li, Y. (2013). Guaranteeing fault-tolerant requirement load balancing scheme based on VM migration. *The Computer Journal, 57,* 225–232.
76. Xiao, X., Zheng, W., Xia, Y., Sun, X., Peng, Q., & Guo, Y. (2019). A workload-aware VM consolidation method based on coalitional game for energy-saving in cloud. *IEEE Access, 7,* 80421–80430.
77. Shukla, R., Gupta, R. K., & Kashyap, R. (2019). A Multiphase Pre-copy Strategy for the Virtual Machine Migration in Cloud. *Smart Intelligent Computing and Applications*. Singapore: Springer.
78. Kapil, D., Pilli E. S., Joshi R. C. (2012). Live virtual machine migration techniques: Survey and research challenges. In *3rd international advance computing conference (IACC)*, IEEE, New York (pp. 963–969).
79. Abe, Y., Geambasu, R., Joshi, K., & Satyanarayana, M. (2016). Urgent virtual machine eviction with enlightened post-copy. ACM.
80. Hu, L., Zhao, J., Xu, G., Ding, Y., & Chu, J. (2013). HMDC: Live virtual machine migration based on hybrid memory copy and delta compression. *Applied Mathematics, 7,* 639–646.
81. Zhu, L., Chen, J., He, Q., Huang, D., & Wu, S. (2013). A smart iteration-termination criterion based live virtual machine migration. *Network and Parallel Computing* (pp. 118–129). Berlin: Springer.
82. Shribman, A., & Hudzia, B. (2013). Pre-Copy and post-copy VM live migration for memory intensive applications. *uro-Par: parallel processing workshops* (pp. 539–547). Berlin: Springer.
83. Nayak, P.C., Garg, D., Shakva, A. (2018). A research paper of existing live VM migration and a hybrid VM migration approach in cloud computing. In *2018 2nd international conference on trends in electronics and informatics (ICOEI).*
84. Yin, F., Liu, W., & Song, J. (2014). Live virtual machine migration with an optimized three-stage-memory copy. *Future Information Technology* (pp. 69–75). Berlin: Springer.
85. Aikema, D., Mirtchovski, A., Kiddle, C., & Simmonds, R. (2012) Green cloud VM migration: Power use analysis. In *International green computing conference (IGCC), IEEE* (pp. 1–6).
86. Zhou, H., Li, Q., Kwang, K.-, & Zha, H. (2018). DADTA: A novel adaptive strategy for energy and performance efficient virtual machine consolidation. *Journal of Parallel Distribution Computing, 121,* 15–26.
87. Sansottera, A., Zoni, D., Cremonesi, P., & Fornaciari, W. (2012). Consolidation of multi-tier workloads performance and reliability constraints. In *International conference presented at the high performance computing a simulation (HPCS),* 2012, Madrid, Spain.

88. Ferreto, T. C., Netto, M. A., Calheiros, R. N., & De Rose, C. A. (2011). Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems, 27,* 1027–1034.
89. Rahman, M., & Graha, P. (2017). Compatibility-base Static VM Placement Minimizing Interface. *Journal of Network and Computer Applications, 84,* 1–21.
90. Beloglazov, A., & Buyya, R. (2011). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic of virtual machines in cloud data centers. *Concurrency and computation: Practice and Experience, 24*(13), 1397–1420.
91. Rai, R., Sahoo, G., & Mehfuz, S. (2017). *Effect of VM Selection Heuristics on Energy Consumption and SLAs During VM Migrations in Cloud. Data Centers. Advances in Computational Intelligence* (pp. 189–199). New York: Springer.
92. Terra-Neves, M., Lynce, I., & Manquinho, V. (2018). Virtual machine consolidation using constraint-based multi-objective optimization. *Journal of Heuristics, 25*(3), 339–375.
93. Arcaini, P., Holom, R.-M., & Riccobene, E. (2016). ASM-based formal design of an adaptivity component for a Cloud system. *Formal Aspects of Computing, 28*(4), 567–595.
94. Ruiz, M. C., Cazorla, D., Pérez, D., & Conejero, J. (2016). Formal performance evaluation of the Map/Reduce framework within cloud computing. *The Journal of Supercomputing, 72*(8), 3136–3155.
95. Abid, R., Salaün, G., & De Palma, N. (2016). Formal design of dynamic reconfiguration protocol for cloud applications. *Science of Computer Programming, 117,* 1–16.
96. De, S., & De, S. (2016). Modeling decoupled mobile cloud computing using mobile UNITY. *Concurrency and Computation: Practice and Experience, 28*(10), 2811–2855.
97. Souri, A., Navimipour, N. J., & Rahmani, A. M. (2017). Formal verification approaches and standards in the cloud computing: A comprehensive and systematic review. *Computer Standards & Interfaces, 58,* 1–22.
98. Sandıkkaya, M. T., Ovatman, T., & Harmancı, A. E. (2015). Design and formal verification of a cloud compliant secure logging mechanism. *IET Information Security, 10*(4), 203–214.
99. Ficco, M., Palmieri, F., & Castiglione, A. (2015). Modeling security requirements for cloud-based system development. *Concurrency and Computation: Practice and Experience, 27*(8), 2107–2124.
100. Jarraya, Y., Eghtesadi, A., Sadri, S., Debbabi, M., & Pourzandi, M. (2015). Verification of firewall reconfiguration for virtual machines migrations in the cloud. *Computer Networks, 93,* 480–491.
101. Rezaee, A., Rahmani, A. M., Movaghar, A., & Teshnehlab, M. (2014). Formal process algebraic modeling, verification, and analysis of an abstract Fuzzy Inference Cloud Service. *The Journal of Supercomputing, 67*(2), 345–383.
102. Deng, P., Ren, G., Yuan, W., Chen, F., & Hua, Q. (2015). An integrated framework of formal methods for interaction behaviors among industrial equipment. *Microprocessors and Microsystems, 39*(8), 1296–1304.
103. Keshanchi, B., Souri, A., & Navimipour, N. J. (2016). An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing. *Journal of Systems and Software, 124,* 1–21.
104. Cao, J.-W., Zhang, F., Xu, K., Liu, L.-C., & Wu, C. (2011). Formal verification of temporal properties for reduced overhead in grid scientific workflows. *Journal of Computer Science and Technology, 26*(6), 1017–1030.
105. Amoretti, M., Grazioli, A., Senni, V., Tiezzi, F., & Zanichelli, F. (2014). A formalized framework for mobile cloud computing. *Service Oriented Computing and Applications, 9*(3), 229–248.
106. Salaün, G., Boyer, F., Coupaye, T., De Palma, N., Etchevers, X., & Gruber, O. (2013). An experience report on the verification of autonomic protocols in the cloud. *Innovations in Systems and Software Engineering, 9*(2), 105–117.
107. Koomosny, D., Mrdovic, S., Ilka, P., Grejtak, M., & Paspichal, O. (2017). Testing Internet applications and services using Planet Lab. *Computer Standards & Interfaces, 53,* 33–38.

Rahmat Zolfaghari is presently working as faculty in Department of Computer Engineering, Hashtgerd Branch, Islamic Azad University (H IAU), Tehran, Iran. He received his B.S. in Software Engineering from Shahid Beheshti University (SBU), Tehran, Iran, the MSc in Software Engineering from Sharif University of Technology (STU), Tehran, Iran. He is a full-time PhD Candidate in Engineering-Software at Science and Research Branch University (SRBU), IAU University, Tehran, IRAN. His research interests are Database, Software design and Modelling, Distributed system, Cloud computing.

Amir Masoud Rahmani received his BS in Computer Engineering from Amir Kabir University, Tehran, in 1996, the MS in Computer Engineering from Sharif University of Technology, Tehran, in 1998 and the PhD degree in Computer Engineering from IAU University, Tehran, in 2005. Currently, he is a Professor in the Department of Computer Engineering at the IAU University. He is the author/co-author of more than 200 publications in technical journals and conferences. His research interests are in the areas of distributed systems, Internet of things and evolutionary computing.