Adaptive Computational Solutions to Energy Efficiency in Cloud Computing Environment Using VM Consolidation

Bhagyalakshmi Magotra · Deepti Malhotra · Amit Kr. Dogra

Received: 11 August 2022 / Accepted: 5 November 2022 / Published online: 27 November 2022
© The Author(s) under exclusive licence to International Center for Numerical Methods in Engineering (CIMNE) 2022

Abstract
Cloud Computing has emerged as a computing paradigm where services are provided through the internet in recent years. Offering on-demand services has transformed the IT companies' working environment, leading to a linearly increasing trend of its usage. The provisioning of the Computing infrastructure is achieved with the help of virtual machines. A great figure of physical devices is required to satisfy the users' resource requirements. To meet the requirements of the submitted workloads that are usually dynamic, the cloud data centers cause the over-provisioning of cloud resources. The result of this over-provisioning is the resource wastage with an increase in the levels of energy consumption, causing a raised operational cost. High CO$_2$ emissions result from this huge energy consumption by data centers, posing a threat to environmental stability. The environmental concern demands for the controlled energy consumption, which can be attained by optimal usage of resources to achieve in the server load, by minimizing the number of active nodes, and by minimizing the frequency of switching between active and de-active server mode in the data center. Motivated by these actualities, we discuss numerous statistical, deterministic, probabilistic, machine learning and optimization based computational solutions for the cloud computing environment. A comparative analysis of the computational methods, on the basis of architecture, consolidation step involved, objectives achieved, simulators involved and resources utilized, has also been presented. A taxonomy for virtual machine (VM) consolidation has also been derived in this research article followed by emerging challenges and research gaps in the field of VM consolidation in cloud computing environment.

1 Introduction
A distributed system is a set of independent elements that work together to accomplish a common goal. As stated by [1], Cloud Computing (CC) can be referred to as a form that originated from distributed computing and has revolutionized the industry of Information and Communication Technology (ICT) by introducing the concept of on-demand availability of computing resources. Because of the significant development in the terms of capabilities of the technology, computational resources have become easily available. This advancement in technology has led to the emergence of cloud computing wherein the resources are provided to multiple users on an on-demand and sharing basis. Cloud Computing is considered one of the most important computing paradigms in IT sector. Based upon the internet technology, the emergence of CC has provided services to the applications that are compute-intensive. To provide the compute resources on-demand over the internet located at a remote data center, the cloud providers share a pool of resources that can be accessed from any location in the world. Applications and data are stored in a data center. The services are provided by the cloud service providers to the users via cloud data centers. The workload of the data centers is heterogeneous and thus, proper provisioning of the resources is required to provide good quality of service to the users. Cloud computing is majorly dependent on its functioning on a technology named virtualization. Underutilization of mainframe computers led to the development of Virtualization technology by IBM in 1960 to make the most out of hardware resources. With the help of virtualization, multiple virtual machines can run over a single host machine. The concept of Virtualization enables the end-users and the service providers to have efficient utilization of cloud resources with
Various objectives like balancing of load, energy management, the sharing of resources among multiple users, making the system fault-free, can be achieved with the help of virtualization [3]. With this advancement in technology, there is a great pace escalation in the number of users requesting resources in CDCs. A great pace escalation in the number of users requesting resources in CDCs has exponentially raised the power consumption making the network operation costly. This ever-increasing process to resource ratio results in degradation of network performance and increased energy consumption. Data centers are one of the major contributors to world’s power and energy consumption. Further, due to the outbreak triggered by the novel coronavirus, the organizations have suspended office work and suggested that employees work from home. This has led to an enormous increase in the use of cloud computing services. It is predicted that the information technology (IT) sector will consume up to 13% of global electricity by 2030, which is currently 7%. This percentage of energy consumption is increasing at the rate of 12% every year. According to the analysis done in 2020, 60% of the total data traffic was consumed by online shopping, gaming, and streaming, which is also forecasted to raise to 80% in the next five years [4]. The increased energy consumption is not only due to the electronic devices involved in the cloud system but also because of inefficient utilization of resources. Improvement in cloud resource utilization can lead to the minimization of this energy consumption. Here, the dynamic VM consolidation comes into play to efficiently reduce the amount of energy consumed and the carbon footprints [5] of the data center. Figure 1 shows the major reasons for energy inefficiency in the data centers. Server utilization is represented in terms of the ratio of the number of resources used to the total resource capacity of the server. For example, the current utilization of the CPU of a machine divided by its maximum capacity of the CPU gives an estimate of the utilization of that particular machine. Over the last decade, the demand for computing resources has increased exponentially. This ever-increasing process to resource ratio has led to various performance issues [6]. The key problem with this growing demand for various cloud resources is the overutilization of resources which results in degradation of network performance.

Also, if the servers remain underutilized, they result in an exceptional increase in operational cost and power consumption. Even an idle server consumes 70% of its peak power [7]. This overutilization and underutilization of resources have resulted in low server utilization, which in turn, is a major reason behind the wastage of energy. This poses a great concern for the management of resources in the data centers. Across the world, many initiatives [8] like Climate Neutral Data Centre Pact and European Green Deal are being taken to make the data centers climate neutral. Numerous schemes have also been proposed and several frameworks have been developed by various authors aimed at augmenting the energy efficiency of the cloud computing environment. However, virtual machine consolidation has proved to be the most efficient of all these techniques owing to its ability to reduce energy consumption while trying to avoid the SLA violations [9].

Apart from energy consumption, some other important challenges in cloud computing include transportability of data over cloud, privacy and secrecy of the shared data, reliability of the services rendered by the providers, computing in low bandwidth environments and the quality of service [10].

1.1 Scope of the Survey

Cloud Computing is considered one of the most important computing paradigms in IT sector. Based upon the internet technology, the emergence of CC has provided services to the applications that are compute-intensive. It plays a very substantial role in this era of digitization. A large figure of surveys has been conducted to analyze multiple facets of cloud computing services. However, the current research focusses on in-depth and detailed review of the consolidation techniques based upon the consolidation step involved, the type of architecture involved, the experimentation environment used, different quality metrics achieved, classification of techniques on the basis of prediction methods employed, incorporation of metaheuristic techniques. The comprehensive survey performed also suggests various challenges and open issues. A taxonomy has also been developed considering the seminal work done by various researchers along with the proposal of a consolidation model. Various state-of-art surveys have been compared in Table 1 on the basis of the above-mentioned factors, which has not been done before.

2 Research Contributions

The key contributions of the current research are as follows:

A taxonomy focussing on various computational engineering-based solutions including statistical and optimization methods for VM consolidation has been presented.
Detailed analysis of the existing consolidation solutions based upon the architecture, predictive methods employed, resources utilized, objectives considered and the consolidation step involved has been presented in the research article. Observing the research gaps, various future recommendations are suggested to ensure energy efficiency in the cloud computing environment. Considering the research gaps, a model for VM consolidation based on has been proposed.

2.1 Methods

An appropriate methodology has been adopted for conducting this survey. For performing the detailed analysis, several research articles have been identified from high quality journals and conferences like IEEE Xplore, Springer, Wiley, ACM and science Direct. For having recent facts and figures, government reports, articles and blogs have also been referred. The keywords like energy reduction, migration, sustainable cloud, energy conservation, efficient cloud data centers, dynamic VM consolidation, overloaded host detection, VM selection and placement have been used for performing the exploratory research.

2.2 Organization

The research article is divided into various sections and the organization of the same is shown in Fig. 2. Section 1 talks about the introduction to cloud computing, the scope of this survey, the contributions of this research article, the methods and materials used for performing this survey followed by the organization of the article. In Sect. 2, we present an overview of different hardware and software centric solutions for reducing energy consumption in cloud data centres, followed by the benefits of VM Consolidation, and the steps involved in the process of consolidation. Section 3 details about various schemes put forward incorporating different consolidation steps. Section 4 includes the analysis of the existing consolidation solution schemes in terms of architecture, resources, objectives and predictive models applied and development of a taxonomy based upon these solutions. The next section provides a glimpse of gaps in the existing literature followed by a proposed framework, with an attempt to overcome few of the challenges. The article has been concluded in Sect. 7.

3 Background

This section primarily focusses on the energy reduction methods that are available in cloud computing. The section is further bifurcated into three sections. The first subsection talks about the different techniques available for energy reduction. The second subsection details about the need of VM consolidation. Whereas, the steps involved in the process of consolidation have been described in the third subsection.
3.1 Power Saving Techniques for CC Environment

In literature, various techniques have been put forward to achieve the goal of energy efficiency in cloud data centres. These techniques have been categorized into two main categories: software centric techniques and the hardware centric techniques. The hardware centric techniques focus on providing the solutions based on infrastructure of the data centres and the IT components involved in it. This includes using variable speed fans, slower drives, cooling equipment etc. However, the software centric solutions make full utilization of the virtualization technology. The categorization of these power saving techniques is shown in Fig. 3.

3.2 Need of VM Consolidation

To deal with the hardware failure of the physical machines, the addition, and deletion of new virtual machines in the cloud resource management system, and to accomplish the task of serving the resource requests that keep evolving, the mapping and remapping of virtual machines across different hosts become unavoidable. **Server Consolidation is a technique that deals with the above-stated situations.** Consolidation aims to minimize the total number of hosts to house a given set of virtual machines, thereby increasing the energy efficiency of the cloud data center. Consolidation of virtual machines is a method of improving resource utilization,
refining energy efficiency, and reducing power consumption. This is achieved by enabling virtualization. Virtualization exploits migration to achieve the following goals:

1. To configure virtual machines on a minimum number of physical machines.
2. To manage overloaded/underloaded hosts.
3. To turn off the idle nodes or keep them at low power hibernation mode.

These goals form the objectives of the dynamic consolidation of machines. With the help of the consolidation of virtual machines, the performance of the system is improved by migrating the virtual machines from one physical machine to another, aiming to balance the load. The intent is the avoidance of the usage of extra physical machines. Virtual machine consolidation is enabled using the technology of VM migration. An example of consolidation is shown in Fig. 4.

Suppose there are 5 physical machines denoted as \( \{PM_1, PM_2, PM_3, PM_4, PM_5\} \) each having the CPU utilization of \( \{50\%, 25\%, 40\%, 30\%, 10\%\} \) respectively. It can be seen that each physical machine has the scope of accommodating more virtual machines since they are not working at their full utilization. By migrating the virtual machines of \( PM_4 \) to \( PM_1 \), \( PM_4 \) can be put to hibernate mode or can be turned off, making \( PM_1 \) work at 80% utilization. Similarly, the virtual machines from \( PM_3 \) and \( PM_5 \) can be migrated to \( PM_2 \). The \( PM_2 \) now works at 75% utilization. Before consolidation, there were 5 active hosts. However, after consolidation, the number of hosts has reduced from 5 to only 2. Migrating the virtual machines between the hosts can reduce the number of active hosts and minimize energy consumption.

### 3.3 VM Consolidation Steps

The consolidation process is triggered by the detection of an overloaded or underloaded machine. The time to begin consolidation is when the host load exceeds the upper threshold or goes below the lower threshold. Once the overloaded or underloaded host machines have been identified, the next step is the selection of one or more virtual machines from the overloaded machines for migration. The virtual machines so selected help to bring back the host load to normal. In the case of underloaded machines, all the virtual machines are selected for migration so that after migration the underloaded machine can be turned off or put to an idle state.
mode to save energy. The final step is the placement of the selected virtual machines to some other physical machine. This deployment step demands the migration of the virtual machines from one host to another. Figure 5 shows the steps involved in the process of consolidation.

4 State of Art for Energy-Efficient VM Consolidation

This section talks about the seminal work of researchers providing solutions for different steps of consolidation. The process of consolidation can be performed in two ways. The first is the static consolidation whereas another is dynamic consolidation. In the case of static consolidation, the virtual machines are configured only once according to the peak demands of the workloads. This ensures that virtual machines do not get overloaded and stay in the same host throughout their lifetime. Static consolidation though avoids machines from being overloaded is also a major cause of the idleness of machines resulting in resource wastage. This problem can be solved using the method of dynamic consolidation. In the case of dynamic consolidation, the capacities of the machines change dynamically according to the demand in workload. The changing demands may call for the migration of virtual machines on finding a host getting overloaded or underloaded. A categorization of the techniques that are involved in the different steps of dynamic consolidation, across different phases, is shown in Fig. 6. First of all, dynamic consolidation checks for any changes in the workload periodically and performs the update accordingly and occurs mainly for two reasons:

(i) When the capacities of the virtual machines exceed the holding capacity of the host.
(ii) When the virtual machines can be accommodated to other hosts to switch the host, they are placed on, to sleep mode.

4.1 Consolidation Time Selection

A dynamic consolidation is prompted whenever a host is detected as overloaded or underloaded. To find the load status, various threshold-based techniques have been suggested in the literature. In the threshold-based techniques, an upper or lower threshold is set. Comparing the utilization values with this threshold value, a machine can be recognized as overloaded or underloaded. The threshold can be either static or dynamic. In the case of a static threshold, the threshold values remain fixed throughout and do not change whereas, with a dynamic threshold, the values can change over a period depending upon certain criteria.
4.1.1 Detection Based Upon Static Threshold

Aimed at minimizing the power consumption without SLA being violated, Beloglazov et al. [19] have proposed a technique based upon the double threshold, upper threshold, and lower threshold, to decide upon the time for the consolidation process to begin. The proposed technique tries to keep the CPU utilization of the servers in between these two defined thresholds. If the utilization of any host exceeds the upper threshold, then VM(s) are migrated from this host to some other host, trying to bring the utilization of the overloaded host below the threshold value. The goal is to avoid SLA violations. On the other hand, if the total utilization value is below the lower threshold, then to reduce the power consumption, the virtual machines from the underloaded hosts can be migrated to other locations. Such migrations can help to turn off the hosting parties and thus, the underloaded machines can be switched to sleep mode from being active. This was followed by a static double threshold method with considerations on CPU and disk utilization [20]. Yahya et al. [21] have focused on setting a threshold based upon the temperature metric of the host, unlike the traditional methods that have focused on other resources of the system like CPU, RAM or bandwidth. A parameter called excess capacity has been coined by the authors [22] to evaluate the remaining capacities. Excess capacity equals the maximum capacity minus the sum of CPU capacities of all the VMs placed on that host. Xiao et al. [23] have focused on the utilization of CPU to set double thresholds to decide upon the load status of the host. The two thresholds are set in the static manner dividing the list of hosts into the categories of overloaded hosts, underloaded hosts, and the normally loaded hosts. A summary of Host load detection techniques based on the static threshold is shown in Table 2.

### Table 2: Static Threshold-based host load detection techniques

| Authors                  | Main idea                                                                 | Objectives                                      | Simulator          |
|--------------------------|---------------------------------------------------------------------------|-------------------------------------------------|--------------------|
| Beloglazov et al. [19]   | Introduced double threshold Lower threshold to find underutilized hosts  | To improve energy efficiency                    | CloudSim           |
|                          | Upper threshold to identify overloaded hosts                              |                                                 |                    |
| Li et al. [20]           | Introduction of double threshold for multiple resources                   | Reduced energy consumption                       | CloudSim           |
|                          | Resources include CPU and Disk                                            | Reduced number of active hosts                   |                    |
|                          |                                                                         | Improved number of migrations                    |                    |
|                          |                                                                         | Improved QoS                                     |                    |
| Yahya et al. [21]        | Use of server temperature for setting the threshold for finding overloading| Reduced energy consumption                       | CloudSim           |
|                          | Categorization of servers to decide upon which one is underloaded          | Reduced SLA violations                           |                    |
|                          |                                                                         | Improved QoS                                     |                    |
| Chinniah et al. [22]     | Hotspot detection on the basis of excess capacity                         | Improved resource utilization                    | Open Nebula        |
|                          |                                                                         | Improve job success rate                         |                    |
| Xiao et al. [23]         | Grouping of hosts into three categories, high, low and normal, based upon double thresholds | Reduced energy Consumption                       | CloudSim           |
|                          |                                                                         | Optimized SLA violations                         |                    |

4.1.2 Detection Based Upon Adaptive Threshold

It has been observed that static thresholds are not appropriate for dynamic workloads. To deal with the dynamic nature of the workloads and changing needs of the users in the cloud system, Beloglazov and Buyya [24] have proposed a few methods based on adaptive thresholds. The main idea behind adaptive thresholds is to evaluate the deviations of CPU utilization. The more the deviation, the more are the chances of utilization to reach full utilization and so, the lesser is the set upper threshold. For the adaptive threshold, statistical analysis of historical data is performed. The approaches include Median Absolute Deviation, Inter Quartile Range, and local regression. To measure statistical dispersion MAD is used. It is defined as the median of the deviations of the utilization from the median of the data. This calculated value helps in calculating the upper threshold. IQR is the difference between the first and third quartiles. The method of local regression suffers from the problem of outliers. To deal with this issue, robust local regression (LRR) has been proposed. To improve the efficiency of the network, Chang et al. [25] have predicted the workload on the servers with the help of recurrent neural networks. To minimize the violations of SLA and performance degradation while maintaining the optimal level of energy consumption, Sharma and Saini [26] have introduced a novel technique for the consolidation of virtual machines in the cloud environment. In 2017, Farahnakian et al. [27] have proposed a self-adaptive algorithm for threshold, detection while considering CPU and memory as resources for calculating the utilization. Depending upon the probability of the predicted error, Minarolli et al. [28] have also taken care of the overheads of live migration. For making predictions, the Gaussian processes have been used whereas the kernel density estimation method has been used for modeling the error prediction. To minimize the energy...
consumption of the cloud computing environment, Damboville et al. [29] have considered the scheduling problem where each server is assumed to have a variable speed and the power of the servers depends upon the processing speeds. Depending upon the deviations of the demands of the available resource, a probability of being overloaded is calculated by Li et al. [30]. This overload probability is used to calculate the adaptive threshold value. Patel and Patel [31] have tried to predict the maximum number of hosts that can be vacated to normalize the host. This predicted value helps in determining the lower threshold. The work proposed by [31] has been improved by Saadi and El Kafhal [32], considering both, the utilization and the characteristics of the server to minimize the overall consumed energy. For selecting an overloaded host, a self-adaptive technique having dynamicity in nature has been put forward by Xie et al. [33]. The proposed technique uses multiple thresholds to check the utilization of the host in the system. In 2018, Zhou et al. [34] have analyzed the application of time series to evaluate ascending and descending trends (ADT) in host overload detection. Table 3 shows an overview of adaptive threshold-based host overload detection techniques. A prediction model based on hybrid recurrent neural network has also been suggested by Karim et al. [35].

4.1.3 Regression-Based Adaptive Threshold

Based upon the prediction of CPU usage, Farahnakian et al. [36] have presented a novel technique based on linear regression to find out the overloaded and underloaded hosts in the cloud environment. The process of host detection using short-term prediction is initiated during the live migration of virtual machines while the process of VM consolidation is being carried out. Given the current resource request, the regression model is used to forecast the near future host utilization. For the detection of overloaded

| Authors            | Main Idea                                                                 | Objectives                               | Simulator |
|--------------------|---------------------------------------------------------------------------|------------------------------------------|-----------|
| Beloglazov and Buyya [24] 2012 | Adaptive threshold CPU utilization deviation dependent MAD method Dispersion based IQR method Half IQR equals MAD in symmetric distribution | Reduced energy consumption Optimized SLA violations | CloudSim |
| Sharma and Saini [26] 2016 | Median based approach Sorting of machines on the basis of utilization history | Minimized performance degradation Reduced SLA violation | CloudSim |
| Farahnakian et al. [27] 2017 | Q learning based threshold detection Inclusion of both memory and CPU usage No need of prior environment knowledge | Minimized number of migrations Minimized energy consumption | CloudSim |
| Minarolli et al. [28] 2017 | Probabilistic approach for finding the future load Use of cumulative distribution function Consideration of uncertainties in predictions | Reduced energy consumption Reduced SLA violation Reduced number of live migrations | CloudSim |
| Li et al. [30] 2017 | Development of estimation model based on Bayesian network Mean and standard deviation have been utilized The probabilistic model has been designed for live migrations | Improved QoS Reduced number of migrations | CloudSim |
| Patel and Patel [31] 2017 | Development of load detection model based on host utilization Determining threshold value using the predicted value of maximum number of hosts that can be released | Improved energy efficiency Improved utilization of hosts Minimized number of migrations | CloudSim |
| Xie et al. [33] 2018 | Dynamic framework, adaptive in nature, capable of predicting and utilizing future utilizations for threshold calculation Consideration of multiple thresholds | Improved QoS Better energy efficiency Reduced number of active hosts | CloudSim |
| Zhou et al. [34] 2018 | Evaluation of ascending and descending trends (ADT) of utilization using forecasting techniques Steepness and extent of ADT determine the threshold which is adaptive in nature | Improved energy efficiency | CloudSim |
| Saadi and El Kafhal [32] 2020 | Determining upper threshold using ratio of performance and power (PPR) Determining lower threshold with the help of overall CDC resource utilization Aimed at making the data center work at its best throughput | Improved energy efficiency Reduced migration count Minimization of violations of SLA Decreased live migration number | CloudSim |
hosts, Yadav et al. [37] have proposed two adaptive threshold models based upon regression. One is named as Gradient descent while the other is maximize correlation percentage-based regression. To optimize the load and the consumed power in the computing environment, Hieu et al. [38] have tried to find out the over utilized and underutilized servers depending upon current resource utilization as well as predicted future utilization of the server. The future utilization is estimated based upon the local history of the resource utilization. However, in 2017, Abdelsamea et al. [39] have considered RAM and BW also along with CPU, extending the concept of linear regression, while computing the host utilization. Yadav and Zhang in [40] have developed M estimation regression (MeReg) where regression is used with heuristics. Jararweh et al. [41] have used a static threshold for determining the lower threshold whereas the upper threshold is calculated using MAD [24]. 10% resource utilization is treated as the lower threshold point. Whenever the utilization of any host falls below this predefined threshold, the host is considered to be underloaded. Apart from energy consumption and SLA violations, the authors [42] have laid stress on the robustness of the upper threshold of CPU utilization which helps in maximizing the usage of the resources. Intending to reduce energy consumption, the authors [43] have focussed on maximizing resource utilization by trying to minimize the degree of imbalance between the hosts. Pearson correlation coefficient has been taken into consideration while finding out the overloaded host. Minhaj Ahmad Khan has also proposed a resource utilization based model for the detection of overloaded hosts [44] while bringing in the concept of cumulative available-to-total ratio for determining the unloaded hosts. In Table 4, regression-based host overload detection schemes have been summarized.

4.1.4 K Nearest Neighbour (KNN) Based Adaptive Threshold

A prediction method based on K Nearest Neighbour has been introduced by Farahnakian et al. [45] to predict future resource usage. as CPU is considered to be the main source of energy consumption, CPU usage has been forecasted. The prediction is dependent upon the historical data which is

| Authors                  | Main idea                                                                                                                                                                                                 | Objectives                                      | Simulator |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|-----------|
| Farahnakian et al. [36]  | Based on linear regression Approximation of a function using utilization history to predict short term future utilization                                                                                 | Minimized power consumption Reduced SLA violations | CloudSim  |
| Hieu et al. [38] 2017    | Regression-based multiple usage prediction Overloaded if, for any resource, both current and predicted usage surpasses the set threshold Underloaded if predicted usage falls below the current usage  | Reduced frequency of migrations Reduced frequency of server switching Better energy saving     | CloudSim  |
| Abdelsamea et al. [39] 2017 | Proposed a regression-based overload detection algorithm Use of multivariate regression Calculation of regression coefficients using Ordinary Least Square Multiple Regression function | Reduced energy consumption                      | CloudSim  |
| Yadav and Zhang [40] 2017 | Regression-based on the median of squared residuals The objective is to minimize the median of the squares value Optimal solution found using Tukey’s bi-square function | Minimized energy consumption Reduced SLA violations Maximized QoS | CloudSim  |
| Yadav et al. [37] 2018   | Development of Gradient Descent (GD) based regression and Maximize Correlation Percentage (MCP) adaptive threshold techniques GD is a machine learning-based MCP is a probabilistic method   | Improved energy efficiency Reduced host reactivations Minimized SLA violations | CloudSim  |
| Jararweh et al. [41] 2018 | Logistic Regression and Median Absolute Deviation based load detection algorithm Inputs to LR are current utilization, utilization history, and upper threshold                                                                 | Reduced energy consumption Reduced SLA violations | CloudSim  |
| Yadav et al. [42] 2020   | Aimed at minimizing residuals All residuals reaching zero point to an optimal solution The goal is to find the slope and intercept values of the regression line Statistical analysis of historical data | Minimized energy consumption Minimized SLA violations Reduced performance degradation | CloudSim  |
| Mapetu et al. [43] 2020  | Calculation of upper threshold using LR and Pearson Correlation Calculation of lower threshold using Current utilization and Pearson Correlation                                                                 | Improved performance Reduced SLA violations Minimized migrations | CloudSim  |
collected during the lifetime of the host as an input training set. Farahnakian et al. [46] have modified the best fit decreasing algorithm to place the migrated VMs. The modification so performed undergoes a prediction of CPU utilization in the early stages that has been done with the help of K nearest neighbor regression. To reduce energy consumption, Zhou et al. in [47] have put forward a dynamic threshold method named as adaptive three thresholds aware technique. The technique makes use of historical data on resource utilization. The hosts have been divided into three categories namely highly loaded, moderately loaded, and lightly loaded. Two adaptive algorithms have been proposed. The two algorithms are K means clustering-based. One combined with MAD(KAM) and the other with IQR(KAI). Table 5 summarises the schemes detecting host overload based on the K nearest neighbor method.

4.1.5 Machine Learning-Based Adaptive Threshold

The CPU utilizations keep on changing frequently depending upon the user’s demands. This leads to unnecessary migrations making any SLA violations. To avoid these frequent migrations, the authors have proposed a double threshold mechanism. For overload detection, Hopcroft et al. [48] have used the concept of the Gray Markov model. The model has been used for prediction purposes. If the current load of CPU utilization is greater than the threshold value, then the next three consecutive CPU utilization are checked against the threshold. If they are also greater than the defined threshold, the corresponding host is said to be overloaded. In 2017, Melhem et al. [49] proposed a method based on present and forecasted loads to check if there is a need to migrate the virtual machines. For the current load, status is checked either using a static threshold method or the dynamic one. In the case of static threshold, the value of the lower threshold corresponds to 0.1 and the upper threshold maps to the value 0.9. However, in the case of the dynamic method the lower threshold is set at 0.1 while for the upper threshold calculation, the Median Absolute Deviation method is employed that utilizes analysis of historical data. Markov chains are used to predict the future load status. The overload threshold has been exhibited as a Markov decision process by Li [50]. In addition to that, the optimization of overload threshold selection has also been performed where the Bellman optimality equation has been used to model the threshold problem. Along with CPU utilization, memory requirements have also been considered to model the problem of consolidation. To address the problem of server consolidation, Hsieh et al. [51] have extended the work of [46] by identifying overloaded and underloaded hosts using the present and the forecasted resource utilization. To forecast the CPU utilization, a Gray-Markov model exploiting the historical data of the hosts has been used by the authors. Instead of servers, the authors have focussed on the future resource utilization of the VM to make more fruitful placements. Markov chains have also been utilized by authors [52] in terms of discrete time to categorize the hosts in terms of their utilizations. In Table 6, host overload detection schemes based upon machine learning have been summarized.

4.2 VM Selection (VMS)

Once the host has been identified for initiating the migration process, the next step is the selection of a VM from the selected host which can be chosen for migration. This sections briefs about numerous such techniques put forward by different authors. Focussing on the problem of the selection of VMs, Beloglazov et al. [19] proposed three policies namely Random Choice (RC), Minimum of migrations(MM), and highest potential growth (HPG) policy. In the random choice strategy, the virtual machine is selected on a random basis, following the uniform distribution. For the minimum of migrations, the minimum number of migrations have opted that reduce

| Authors | Main idea | Objectives | Simulator |
|---------|-----------|------------|-----------|
| Farahnakian et al. [45] 2013 | Prediction based method. The host is underloaded if the total of current and predicted usage is less than 10% of the total capacity K-Nearest Neighbour Regression is used for utilization prediction | Reduced energy consumption Improved SLA | CloudSim |
| Farahnakian et al. [46], 2016 | Current capacity exceeding total capacity and predicted capacity exceeding remaining capacity are the two overloading criteria Use of linear regression and K-Nearest Neighbour regression for prediction | Improved energy efficiency Minimized number of migrations | CloudSim |
| Zhou et al. [47] 2016 | Development of two overload detection algorithms based on K – means clustering One combined with MAD and another with IQR Defined Three thresholds for both KAM and KAI | Improved energy efficiency Fewer migrations Low SLA violation | CloudSim |
the utilization of CPU resources. The HPG policy prevents
SLA violations by selecting the VM with the minimum
usage of CPU. Two more strategies for VM selection have
been proposed in [24]. The MMT policy helps select the V
that requires the minimum time for migration. The other is
the policy of maximum correlation (MC). The maximum
correlation policy finds the virtual machine that shows the
maximum correlation of CPU with other machines. The idea
behind this is that the virtual machine showing higher cor-
relation is more likely to cause the overload in the host. In
maximum correlation, the associativity of a virtual machine
with others is calculated and the machine with maximum
correlation is selected for migration. The authors [53] have
proposed a technique extending the existing Maximum Cor-
relation Policy. They have used the concept of negative and
positive correlation. According to the authors, the virtual
machines having positive correlation are more likely to
cause load imbalance than the virtual machines with nega-
tive correlation. One such machine that has a maximum
positive correlation with others is then chosen for transfer.
Alboaneen et al. in [54] have focussed on the bandwidth of
the available VMs. The VM with maximum requested band-
width is migrated to another host to reduce the load of the
machine on which it is hosted. To select the virtual
machine(s) for migration, Hopcroft et al. [48] have consid-
ered utilized CPU and utilized memory. Depending upon the
contributions made by the CPU of the virtual machines to
the total utilization of the host, Masoumzadeh and Hlavacs
[55] have proposed a policy called the maximum utilization
policy to select the virtual machine with maximum utiliza-
tion as compared to the rest of the virtual machines. Con-
trary to this, a minimum utilization policy has also been
proposed that selects the virtual machine with minimum
utilization. Based upon these two criteria, the virtual
machine selection is performed over the overloaded hosts.
To select the virtual machine for migration from one heavily
loaded host to another, Zhou et al. [47] have proposed three
policies for the VM selection namely Minimum Memory
Size (MMS), Lowest CPU Utilization (LCU), and Minimum
product of both CPU usage and Memory Size (MPCM). In
MMS, the virtual machine with minimum allocated memory
is selected for migration as it will consume lesser bandwidth.
In LCU, the VM with minimum CPU utilization is migrated.
The product of CPU utilization and allocated memory is
calculated in MPCM and the machine with the minimum
value of this product is selected for migration. In 2016, Bala
and Chana [56] have used the correlation factor for VM
selection. In the case of overloaded hosts, the correlation
factor between the virtual machine and the resource utiliza-
tion is calculated. The virtual machine with the maximum
correlation factor is selected for further consideration. On
reaching zero, the VM migration takes place. Taking the
advantage of intelligent decision-making, Alaul et al. [57]
have proposed a technique for VM selection based on fuzzy
logic. To deal with the dynamic nature of the cloud data
center, the authors have provided different kinds of inputs to
the fuzzy system to achieve a better trade-off between the
consumed energy and the SLA violations. These include the
migration time, the correlation factor, and the fluctuations
in CPU utilization. The linguistic variables used for the
inference engine include RAM, correlation, and the standard
deviation. The authors have chosen the trapezoidal member-
ship function as it deals better with the flat regions. A virtual
machine with the highest fuzzy output will be selected for
migration if its CPU utilization exceeds the defined thresh-
old. Shidik et al. [58] have considered RAM along with the
CPU utilization while formulating the VM selection prob-
lem. The problem has been resolved using fuzzy logic and

### Table 6 Machine learning-based host detection techniques

| Authors                  | Main idea                                                                 | Objectives                                | Simulator |
|--------------------------|---------------------------------------------------------------------------|------------------------------------------|-----------|
| Hopcroft et al. [48] 2014| Use of Gray forecasting model High precision modeling with convenient operational execution | Reduced migration frequency Minimized energy consumption | CloudSim |
| Melhem et al. [49] 2017  | Designed a static and a dynamic load detection algorithm Use of first-order Markov chain for static analysis Load prediction using MAD based Markov Model for dynamic detection | Reduced SLA violations Reduced number of migrations | CloudSim |
| Li [50] 2019             | Adaptive overload threshold detection using Markov Decision process Optimization using Bellman Equation | Improved resource utilization Increased energy efficiency | CloudSim |
| Hsieh et al. [51] 2020   | Combined use of Gray model and Markov Chain as GM (1,1) Gray Markov Prediction model Incorporation of Accumulated Generating Operation for noise removal | Reduced energy consumption Reduced SLA violations Prevention of repeated migrations | CloudSim |
| Sayadnavard et al. [52] 2022 | Host status detection based on continuous time and discrete time Markov chains | Reduced energy consumption Avoided VM Migrations | CloudSim |
the Markov algorithm. Fuzzy logic helps in categorizing the attributes of the candidate virtual machines whereas the Markov model is further employed to find out the best VM according to these attributes. The Triangular membership function of fuzzy logic has been used by the authors in their proposed technique. The fuzzy logic helps in simplifying the production rules for the Markov algorithm to reach an optimal decision. The technique has been able to reduce the energy consumption of the cloud data center. Once an overloaded host has been found out with the help of Pearson correlation, the next step is to find out the virtual machine that is to be migrated. To figure out this issue, the authors [59] have used CPU usage and the allocated memory. According to them, the greater the CPU utilization of a VM, the more amount of processors will be released after its migration. Similarly, the lesser the allocated memory, the lesser is the dependence of the virtual machine on others, the lesser will be the migration cost and the time incurred for migration. To find the potential virtual machine for the migration purpose, Hieu et al. [38] have introduced the concept of temperature ratio. The resource with maximum utilization is considered the hottest resource of the host. In 2017, Mosa and Sakellariou [60] have selected the virtual machines with minimum RAM demand so that a minimal amount of is time required for migration. On contrary, Yadav et al. [61] have performed the selection process based on CPU usage. The virtual machines are sorted in descending order of their CPU utilization. The machine with the minimum size is then selected. Thus, the approach selects the machine with high CPU usage and minimum size. This helps in reducing the amount of energy consumed during VM migration. Yadav et al. [37] have utilized the network bandwidth to find out the virtual machine to be selected for migration. The purpose is to select the VM with the least current utilization and MMT. Thus, out of all the virtual machines in the overloaded host, the one that requires minimum time for a migration is selected for migration. Chang et al. [62] have calculated the impact of the virtual machines on the load of the host. This has been achieved by using the Euclidean distance of the resources. The values of the utilization of the resources have been normalized first, which are then used to calculate the Euclidean distance. Wang and Tianfield [63] have considered CPU utilization while selecting the VM. This has been achieved depending upon the CPU utilization of the VMs. The one with maximum utilization is chosen. If the host remains overloaded, then the virtual machine with the highest CPU utilization is selected next to be migrated. This process is repeated until the host load comes below the threshold value. According to the authors, migrating the VM with the highest CPU utilization provides an opportunity to reduce the load very quickly. Li et al. [64] have presented a VM selection technique based on the concept of content similarity. According to the authors, migrating the virtual machines with high content similarity among their memories will help to save the bandwidth to transfer the data along with the reduction in the amount and data transfer time. Lin et al. [65] have built a reverse selection method while considering all the resources i.e., memory, bandwidth, CPU, and storage. The technique detects the most fitting VM from the set of virtual machines for each randomly selected host. This reverse selection depends upon an iterative algorithm that is dependent on the budget heuristic. To achieve better migration costs, Xie et al. [33] have proposed a VM selection algorithm that takes as input all the available resources instead of considering only CPU. The utilization of these resources are used for the calculation of a variable called the utilization ratio of resources. This parameter gives an estimate of the utilization of all the resources (CPU, RAM, BW). The VM with a higher weight is given more preference over others for migration. Zhou et al. [34] have proposed a memory and CPU usage-dependent technique for selecting the migrant virtual machine. In the case of an overloaded host, the virtual machines are ascendingly arranged and the virtual machine having less usage of memory is preferred over other virtual machines. Choudhary et al. [66] have proposed the concept of fuzzy logic into the process of virtual machine selection process. The inputs to the fuzzy logic constitute the ram utilization, the standard deviation of CPU utilization along correlation coefficients. The SLAV parameters have also been added to the list. All these parameters are fed to the fuzzy function. The virtual machine with the highest fuzzy output is selected for migration if its current CPU utilization exceeds the threshold value. The use of fuzzy logic gives better results as compared to other traditional methods of VM selection. Xiao et al. [23] have introduced three different CPU utilization-based virtual machine selection policies after the host has been identified as overloaded. However, if a host is found to be underloaded, then all of its virtual machines are migrated to some other host. Yadav et al. [42] have calculated the CPU utilization of virtual machines over different time frames. The prediction of MAD values has been used to find suitable virtual machines for reallocation from the overloaded hosts. The predicted values help in finding out which virtual machines will put a lesser load as compared to others. Thus, the virtual machine with a minimum MAD value is selected for migration. To select the virtual machine, Saadi and El Kafhali [32] have calculated the deviation of current host utilization from the upper threshold. The virtual machine having the CPU utilization greater than or equal to this deviation is migrated. The machine selected is expected to have the least migration time. Taking imbalance degree into consideration, Mapetu et al. [43] have proposed an optimization policy to reduce the migration count. The selection
criteria are based on the result of the product of maximum MAD and least imbalance degree. Migration delay has been computed with the help of RAM and BW. Table 7 summarizes VM selection policies proposed by numerous authors in the literature.

### 4.3 VM Placement (VMP)

Once the migration virtual machine has been selected, the final job is to find a suitable host that can house this migrated virtual machine. The methods used for finding a host for the deployment of the virtual machine include Greedy Heuristics, Constraint Satisfaction Programming, Linear Programming, and Meta-Heuristic Techniques.

#### 4.3.1 VMP Based on Greedy Heuristics

Conventionally, in the Bin Packing Problem, some specific sized objects are to be packed into bins of fixed sizes. The primary goal is to reduce the total number of bins that can be used to hold the given set of items. Virtual machine placement problem (VMPP) can be modeled as the Bin Packing problem (BPP) where the number of available hosts corresponds to the total bin count while the virtual machines that have been selected for migration can be thought of as the objects of variable sizes that need to be packed in the bins. The Bin Packing problem belongs to the category of NP-hard problems. Mishra and Sahoo [67] studied the use of various greedy algorithms and have put forward the shortcomings with various placement algorithms. Trying to overcome these drawbacks, the authors have tried to represent the placement problem with the help of a mathematical model. Vector arithmetic has been used to represent the VM placement as an object in two-dimensional space with the resources as the dimensions of the object. Motivated by greedy algorithms, the authors [68] have proposed an algorithm called the SERCON. The algorithm inherits the usefulness exhibited by First Fit and Best Fit. The concept behind the proposed scheme is that the least loaded hosts should free their capacities if other hosts can accommodate the virtual machines placed on the existing low-loaded hosts. To achieve this, the very first step is the decreasing sorting of the host machines based on their loads. Once the hosts have been sorted, the virtual machines are selected from the least loaded host based on the calculated scores. For the calculation of scores, the authors have considered CPU and memory resources. In an attempt to reduce the amount of energy consumed, Anton et al. [19] have presented an energy-aware VM placement technique. The proposed scheme is a modification of BFD. The idea behind the proposed work is to choose the host which leads to a minimal rise in the level of power consumption after the placement of the virtual machine. Therefore, the most power-efficient host is chosen first. Huang and Tsang [69] have modeled the virtual machine consolidation as the minimum cost optimization problem that aims to place the virtual machines on a given group of hosts while reducing the service level agreement violations. The problem has been simplified by fixing the number of hosts in advance. This sub-problem has been modeled as the M-convex problem and to solve the problem, the lagrangian method has been used. Fu and Zhou [70] have proposed a scheme for virtual machine placement that offers to compute correlation coefficient during the consolidation process. The correlation coefficient helps in finding the associativity between two objects. The objects mentioned referring to hosts and virtual machines in this case. Thus, the host has a minimum correlation with the migrated VM is chosen for placing it. Esfandiarpoor et al. [71] have considered the network structure and the cooling aspect of the data center for the VMP during the process of consolidation. This reduced the rack count and routers in active mode. The proposed scheme, OBFD, is a variation of modified BFD. Instead of sorting the virtual machines in terms of CPU utilization, they have been arranged in decreasing the order of the requested MIPS. Also, three structure-based virtual machine placement algorithms have been presented to reduce the racks in active mode. To avoid violations of SLA and frequent migrations, Farahnakian et al. [46] have put forward two constraints that need to be checked while migrating the virtual machines. According to the first constraint, the host selected for the placement of the virtual machine should have sufficient resources to accommodate it. The second constraint says that the host should not get overloaded in near future, provided the host holds sufficient resources to cater to the demands. Mosa and Paton [72] designed VMPP with the help of utility functions. The proposed placement strategy helps in optimizing both energy and service level agreements. Intended to benefit the IaaS provider by maximizing the utilization of the available resources, the scheme uses a genetic algorithm to find out the possible solutions to the placement of a VM to a host. The initial population has been generated on a random basis. However, the crossover operation has been performed using the uniform crossover technique where swapping is performed beyond the crossover point. The proposed utility-based method can be applied only if there is an adaption that can be improved for the current allocation. Zhou et al. [47] have extended the best fit decreasing algorithm and have presented its modified version named as Energy-Aware Best Fit Decreasing (EABFD). If the host has sufficient resources to place the migrated machine, then the utilization of the host after the placement of the machine is checked. If the utilization lies between the limits of moderate loaded and lightly loaded machines, then the host is added to the list of considered hosts. All these hosts are then checked for the raised power consumption after the placement of the
| Authors                          | Main idea                                                                 | Objectives                                      | Simulator         |
|---------------------------------|---------------------------------------------------------------------------|-------------------------------------------------|-------------------|
| Beloglazov et al. [19] 2012     | Utilization based three virtual machine selection policies have been proposed | Reduced Energy consumption                      | CloudSim          |
| Beloglazov and Buyya [24] 2012 | Two policies for VM selection have been proposed                           | Reduced SLA violations                          | CloudSim          |
|                                 | MMT is based on migration time                                            | Minimized number of migrations                  |                   |
|                                 | MC uses multiple correlation coefficient                                   | Reduced energy consumption                      |                   |
| Cao [53] 2012                   | Use of positive correlation only for finding virtual machines for selection | Reduced energy and SLA violation                | CloudSim          |
| Alboaneen [54] 2014             | VM with maximum bandwidth request is chosen                                 | Minimized number of migrations                  | CloudSim          |
| Hopcroft et al. [48] 2014       | The selection of VM depends upon lower CPU threshold value, currently occupied memory, and the threshold value of memory utilization ratio | Elimination of frequent migrations              | CloudSim          |
| Masoumzadeh and Hlavacs [55] 2015 | Selection of Virtual Machines based on maximum and least utilizations      | Improved Energy SLA Violation                   | -                 |
| Zhou et al. [47] 2016           | Three policies based on RAM utilization, CPU utilization, and product of both, have been proposed | Improved energy efficiency                      | CloudSim          |
| Alau et al. [57] 2016           | Intelligent Fuzzy logic based VM Selection Algorithm                       | Reduced number of migrations                    | CloudSim          |
|                                 | Migration control strategy incorporated with Fuzzy Logic                   |                                                  |                   |
|                                 | VMs with low CPU usage and high fuzzy output are given preference          |                                                  |                   |
| Shidik et al. [58] 2016         | Application of Markov Normal Algorithm (MA)                               | Reduced SLA time per active host                | CloudSim          |
|                                 | Production rules of MA simplified by Fuzzy Logic                           |                                                  |                   |
| Rajabzadeh and Toroghi [59] 2017 | Virtual machines with high CPU usage and Low memory utilization are selected | Improved energy-SLA trade-off                   | CloudSim          |
| Hieu et al. [38] 2017           | Intended to select the virtual machine that minimizes the resource temperature | Reduced number of migrations                    | CloudSim          |
| Mosa and Sakellariou [60] 2017  | Machine with minimum RAM demand                                            | Minimized migration time                        | CloudSim          |
| Yadav et al. [61] 2017          | Virtual machine with high CPU utilization and minimum size                 | Optimized energy utilization                    | CloudSim/Real workload |
| Yadav et al. [37] 2018          | Selection of virtual machine with minimum migration time                   | Minimized SLAV                                   | CloudSim          |
| Chang et al. [62] 2018          | Residual available capacity based selection has been proposed              | Reduced energy consumption                      | CloudSim/ Real workload |
| Wang and Tianfield [63] 2018    | Selection of virtual machine with the highest utilization until host becomes normally loaded | Assured SLA                                     | CloudSim          |
|                                 |                                                                           | Improved energy efficiency                      |                   |
| Li et al. [64] 2018             | Content-based VM selection VMs having maximum memory content match are selected | Reduced memory pages transfer                   | CloudSim          |
| Xie et al. [33] 2018            | Descending order sorting of calculated weights                             | Reduced migration time                          | CloudSim 4.0      |
|                                 | Selection of VM with the highest weight                                    |                                                  |                   |
| Zhou et al. [34] 2018           | Less memory usage                                                         | Reduced energy consumption                      | CloudSim 4.0      |
|                                 | More CPU usage                                                            | Optimized SLA Violations                        |                   |
| Choudhary et al. [66] 2019      | Fuzzy based VM selection                                                  | Minimized migration time                        | CloudSim 3.0.3    |
|                                 | Use of Trapezoidal membership function                                     |                                                  |                   |
migrated virtual machine. The host with minimal power consumption increase is selected for the deployment of the migrated virtual machine. The proposed framework VMCUP-M by Hieu et al. [38] is an extension of the PABFD algorithm. Where the authors considered multiple resource usage. This multiple resource usage is the predicted value which helps in reducing the chances of the target host being overloaded in the future. The forecasted utilization value and the power increased after the placement of the virtual machine act as the parameters for the selection of the host. The host with a minimal increase in power consumption is selected as the target host. The hidden Markov model has been used by Hammer et al., 2017 [73] to tackle the VMPP. In the proposed scheme, the Hidden Markov Model helps in replicating the consumption of the CPU and its properties in a realistic way. The time of the days decides upon the changeover of the CPU state from being highly active to inactive. Markov model has also been used by Rajabzadeh and Toroghi in [59] for solving VMPP. The presented scheme is an extended version of PABFD [24] with the incorporation of simulated annealing in the Markov model. The deployment of VMs is performed in such a manner that does not leave the selected host overloaded after migration.

With the use of interquartile range, the underloaded and the overloaded hosts will act as outliers, leaving the set of remaining hosts that can be used for the placement of the virtual machines. For these candidate hosts, an energy factor called energy expenditure is computed. The lesser the value of this parameter, the more are the chances of the corresponding host getting selected as the destination for the VMP. To maintain the balance between energy, QoS, and temperature, Zahedifard et al. [74] have introduced a technique that takes care of virtual machine placement during the consolidation process. The hosts have been classified as high category and low category. Some servers consume more energy while performing fewer operations per second. Such servers fall into low category servers. The aim is to prioritize the high category hosts while trying to minimize the low category hosts’ inactive state. Out of the list of high category hosts, the host with minimum power increases and more resource capacity after the placement of the virtual machine is selected for the deployment process. The hosts with more resource capacity are expected to contribute less to frequent migrations. Patel and Patel [31] have presented a utilization aware selection and deployment scheme from the underloaded hosts. The over-utilized and the turned-off hosts are omitted from being selected as the candidate hosts for new virtual machine placements. This leads to reduced computation costs. Motivated by the work of [24], Wang and Tianfield [63] have proposed a technique called Space Aware Best Fit Decreasing. In the suggested scheme, the VMs that need to be migrated are sorted in decreasing order of their CPU utilization. Thus, the VM having maximum utilization would be the first one to get migrated. Three correlation-based algorithms have been presented by Chen et al. [75] to generate an effective placement scheme. Neural networks have been employed to predict resource utilization based upon historical data. This predicted value is then fed to the placement algorithms. The first proposed scheme is a modification of first fit and authors have termed it as correlation aware first for virtual machine placement algorithm. The second algorithm that has been proposed is a modification of the best fit VMP algorithm. The third one is the Minimum Bin slack algorithm. Arani [76] has presented a learning-based placement technique based on Best fit decreasing algorithm. For learning purposes, the authors have used correlation coefficient along with ensemble prediction and automata theory. Incorporating these three with the best fit decreasing algorithm makes the decision-making better. Gupta and Amgoth [77] have proposed a resource-aware scheme to reduce the count of active hosts. The presented scheme is completely resource-aware and deploys the VM on the host with the highest utilization. To target such a kind of placement, the authors have put forward a parameter called Resource Usage factor, RUF. RUF is dependent upon average resource utilization, the normalized value of the resource utilization and the normalized value of the remaining resources. Higher the value of RUF means higher are the
chances of the host of being available for placing the migrated virtual machines. The approach proposed by Xie et al. [33] for VMP is an advancement of the Best Fit Algorithm. In the proposed multi-weight best-fit algorithm, the deviation of the threshold from the actual utilization of the resources, i.e., CPU, RAM, and BW, is computed. The host with a minimal increase in power usage is chosen for housing the migrated virtual machine. Mishra et al. [78] have presented an energy-efficient scheme for the placement of virtual machines. In the proposed scheme, the virtual machines are first divided into different categories depending upon their resource requirements. These categories include CPU intensive, input–output intensive, memory-intensive, and bandwidth-intensive. The host with minimum remaining capacity is the first one to be checked. If no host in active mode can house the virtual machine, then one of the hosts from the list of hosts in sleep mode is made to wake up and the virtual machine is deployed in there. Haghshenas and Mohammadi [79] have presented two approaches based on linear regression to find the suitable destination host. The first technique called the OLLRBA, the One-level linear regression-based approach, and TLLRBA, the Two-level linear regression-based approach, used linear regression at two different points. Mapetu et al. [43] have proposed a Pearson-based PABFD algorithm for seeing into the virtual machine placement problem of the consolidation process. The Pearson correlation coefficient uses the RAM and CPU utilization. The VM is housed by the server which shows the least increase in the power consumption difference. The authors in [32] have extended the work of [31]. The authors stated that the association of the power efficiency and CPU utilization differs from server to server depending upon their technical specifications. Hosts operating at the same CPU utilization may differ in their power consumption. Thus, the type of the server has also been taken into consideration while calculating the CPU utilization. Zang et al. [9] has put forward a reinforcement learning based technique for machine learning. VM placement techniques based on Greedy Heuristics have been summarized in Table 8.

4.3.2 VMP Based on constraint Satisfaction Problem

The aim of the Constraint Satisfaction Problem (CSP) is to evaluate the value for a variable from a given domain of values while satisfying a set of constraints. The problem of VMP can also be modeled as CSP. CSP can be thought of as logical programming which is mainly used to solve complex combinatorial problems. Dupont et al. [80] suggested an energy-aware algorithm for the reallocation of virtual machines. The disassociation of the constraints from the algorithm makes it flexible nature. The objective is to minimize energy consumption and CO2 emissions. Zhang et al. [81] have presented a constraint programming-based allocation model that is capable of handling different types of workloads like CPU intensive or memory-intensive or I/O intensive. At the local level, the utilizations are evaluated to find out the load on the host whereas the second level corresponds to the global functioning of packing the virtual machines on suitable hosts. Tchana et al. [82] have introduced software consolidation into the dynamic VM consolidation. The proposed scheme is based on constraint programming and has been used for live migration where the goal is to reduce energy and save operational costs. The authors have relied on the consolidation of the software which can be used in both public as well as a private cloud. Table 9 summarizes the constraint satisfaction problem-based VM placement techniques.

4.3.3 VMP Based on Linear Programming (LP)

Unlike logical programming, linear programming utilizes estimation models for finding the results with the help of probability distributions. They majorly cover the area of solution finding where the future demands are not certain. Thus, in the cases where the future demands of the machines for resources are not known or certain, the concept of linear programming can be used. Tseng et al. [84] have presented a virtual machine placement as the service-oriented optimization scheme based on integer linear programming. Based on graph theory, a tree and a forest algorithm have been proposed to deal with this optimization problem. Zeng et al. [85] have proposed a network-aware VM placement technique where they have formulated the problem in terms of linear programming. The requirement of the VMs in terms of resources, the traffic between the VM pair, and the rack-wise architectural description of the network form the input to the linear programming. The VMs with high traffic rates are given priority in the sense that, the VMs with heavy traffic are paired to be placed onto the same server. Huang and Tsang [86] have proposed a framework that automates the process of VM consolidation intending to improve the allocation of virtual machines and servers. M convexity theorems and functions have been used to find the solution. Table 10 summarizes linear programming-based VM placement techniques.

4.3.4 VMP Based on Meta-heuristics

Being NP in nature, the problem of VM placement can be modeled as an optimization problem. To handle and solve the optimization problem, various meta-heuristic techniques algorithms have been focussed upon in the literature. Particle Swarm Optimization has been used by various authors to
### Table 8 Greedy heuristics-based VM placement techniques

| Authors                  | Main idea                                                                 | Objectives                                      | Simulator          |
|--------------------------|---------------------------------------------------------------------------|-------------------------------------------------|---------------------|
| Mishra and Sahoo [67] 2011 | Based on vector arithmetic. Representation of resources in 3-D resources. Total capacity vector, total remaining vector, and total imbalance vector define vectors of the cube. | Energy efficiency.                              | XEN                 |
| Murtazaev and Oh [68] 2011 | Inherited from First Fit and Best Fit. VMs from least loaded hosts are migrated to least loaded ones. Rank-based migration. The surrogate weights method has been used for ranking the hosts. | Minimized number of migrations. Minimized number of active hosts. | .NET 3.5 Framework |
| Beloglazov et al. [19] 2012 | Modification of Best Fit Decreasing. Host selection based on least power increase. | Reduced energy consumption.                     | CloudSim            |
| Huang and Tsang [69] 2012  | Defined as a Non-linear programming binary problem. Use of Lagrangian relaxation. Highly scalable. | Reduced SLA violations.                         | –                   |
| Fu and Zhou [70] 2015     | Correlation-based placement. Selection of host-based on a minimum correlation between VM and host. | Improved energy consumption. Reduced SLA violations. | CloudSim 3.0        |
| Farahnakian et al. [46] 2016 | Current capacity and predicted capacity constraints have been used. Accommodation of the VM in the selected host in current as well as future times. | Reduced energy consumption. Reduced SLA violations. Reduced migrations. | CloudSim             |
| Mosa and Paton [72] 2016  | Utility function-based placement. Aimed at maximizing the profit of the cloud provider and minimizing the cost payable by the cloud provider to the user for SLA violations. | Optimized energy consumption. Reduced SLA violations. Lesser VM migrations. | CloudSim             |
| Zhou et a. [47] 2016      | Three threshold-based policy. Place the VM if after allocation the utilization is in between moderate and low levels. Selection of Host with minimum power consumption. | Improved energy efficiency.                     | CloudSim             |
| Hieu et al. [38] 2017     | Prediction based placement technique. PABFDF combined with multiple usage prediction. Incorporation of Utilization stability. Decreased chance of future host overload. | Reduced energy consumption. Reduced number of active servers. Fewer migrations. | Cloudsim/real workloads |
| Hammer et al. [73] 2017    | Forecasting of CPU utilization. Use of stochastic models. Implementation of Hidden Markov Model. Use of Auto-regression for CPU utilization. | Reduced migrations to a great extent. –         | –                   |
| Rajabzadeh and Toroghi [59] 2017 | Use of Markov Chains. Embedded parallel Simulated Annealing in Markov Chains. | Better energy – performance trade-off.          | CloudSim             |
model the problem by VM placement. Ramezani et al. [88] solved the VMP problem using PSO along with the fuzzy logic. The objective is to minimize power consumption while improving resource utilization and transfer time. Resource utilization has been improved by minimizing both, the idle memory and the idle CPU. The fuzzy logic...
has been used for controlling the inertia weight in the basic PSO. With the help of the presented scheme, the network traffic has been greatly reduced. While trying to minimize the energy consumption, Hongjian Li et al. [20] also mapped the virtual machine placement optimization problem to particle swarm optimization. To avoid getting trapped into local optima, the authors have incorporated probability findings based upon the Bayesian formula to find the global best and local best positions. Abdessamia et al. [89] have evaluated three inertia factors based upon the local and best positions. These calculated values help in finding the optimal solution and correspond to the probabilities of selecting the host with maximum energy consumption. In 2018, Tripathi et al. [90], have used binary PSO to optimize VM placement by taking energy usage and utilization of resources into consideration. The objective is to lessen energy consumption while maximizing resource utilization. Trying to minimize the number of hosts in active mode, Yan et al. [91] have concentrated on memory and CPU utilization as parameters of the objective function in discrete PSO. OpenStack has been used to evaluate the capabilities of the proposed algorithm and has been compared with the native VM scheduler of OpenStack. Another PSO based placement technique has been proposed by Kirana and Mello [92] for consolidating the virtual machines, trying to improve the energy efficiency. Several ant colony-based virtual machine placement algorithms have been proposed by different authors. In 2013, Gao et al. [93] proposed a technique to find a non-dominant set of solutions using the Pareto dominance concept, aimed at reducing power consumption and resource wastage. To fully utilize a host, the resource wastage should be minimum. Motivated by [93], Malekloo and Kara [94] have proposed an algorithm combining the properties of heuristic information and probability-based decision rules, with ant colony optimization. The energy consumed by the network elements has been computed. K- shortest path algorithm has been used to find the number of network elements between two VMs. Shabeera et al. [95] have optimized the virtual machine allocation by calculating the distance between two physical machines. The distance is measured in terms of network devices that have been used to connect them. The objective is to minimize the delay between these devices. To deal with the virtual machines having different arrival times and having variable requirements at different time slots, Liu et al. [96] have introduced an energy-aware scheme with multi-objective optimization. A research project is done by Ashraf and Porres [97] employing an ant colony system. Carrying forward the work done by [96], Xiao et al. [23] have also tried to optimize energy and the number of migrations using an ant colony system. However, the authors have put forward a different energy model that considers the energy consumed by the host not only in active mode but also in sleep mode as well as while switching between these modes. Alharbi et al. [98] have presented an ACO-based VM placement technique for reducing the amount of consumed energy. The proposed scheme is profile-based and thus uses the information from the profiles of virtual machines and host machines. The profile of the host included information like host CPU and memory utilization, their total and remaining capacities, the energy consumed at peak and idle times. In recent years, the dragonfly algorithm has been used for optimization in VM consolidation. In 2018, More and Ingle [99] have proposed a multi-objective dragonfly algorithm for the placement of VMs during the consolidation process. The optimal solution has been found by incorporating crow search into a dragonfly. Keeping in view better resource utilization, Tripathi et al. [100] have tried to reduce the resource wastage for the VM deployment using dragonfly optimization. The objective is the minimization of the CPU and memory wastage. To maintain a balance between exploration and exploitation, a time-dependent transfer function has been incorporated into the binary dragonfly. Goyal et al. [101] has presented a whale optimization-based resource allocation scheme for making the cloud environment more energy efficient. Researchers [102–106] have also employed a Genetic Algorithm to find the solution to VM Placement. For the placement of VMs in geographically distributed data centers, Teyeb et al. [107] have put forward a traffic-aware Bio Geography-based optimization-based scheme. The main idea behind the proposed technique is to reduce the number of migrations.

Table 10  LP-based VM placement techniques

| Authors          | Main idea                                                                 | Objectives                                      | Simulator         |
|------------------|---------------------------------------------------------------------------|------------------------------------------------|-------------------|
| Tseng et al. [84]| Service-oriented framework Based on Integer Linear Programming Formulation of two proposed, the Tree and the Forest, algorithms based on graph theory | Minimized communication cost                   | Dev 5 C++         |
| Zeng et al. [85]| Heuristic algorithm for integer programming with quadratic constraints | Lower communication cost                        | Gurobi Optimizer [87] |
| Huang and Tsang  | Use of Kernel Density estimation model M-convexity function has been incorporated Intelligence-based highly configurable algorithm | Improved utilization Reduced number of active servers Reduced number of migrations | CloudSim          |


Due to the addition and removal of virtual machines, there exists an imbalance of load in the host machines. To deal with this imbalance degree, Li et al. [108] have proposed a multi-objective rebalance solution aimed at maintaining the intra-host and inter-host load imbalance of each dimension. Dimension corresponds to different resources of the host. Migration cost has been effectively reduced by the presented scheme. Medara et al. [109] has introduced water wave optimization for VM consolidation. Meta-Heuristics-based VM placement techniques have been summarized in Table 11.

5 Consolidation Based Solutions for Energy Efficiency in CC Environment

This section provides a summary of different VM consolidation techniques discussed in the previous section. The proposed schemes have been classified on the basis of different parameters and thereafter a taxonomy has been developed providing the solutions to the process of consolidation.

5.1 Analysis of Different Consolidation Schemes

Tables 12 and 13 have been formulated after analyzing different consolidation schemes proposed by various researchers over the past few years. They provide the summary of various dynamic virtual machine consolidation schemes studied in Sect. 3. The classification has been done based on the deployed architecture, prediction method used, the step of consolidation involved, the resources used and the objectives achieved.

6 Discussion

With the help of the conducted survey, we have been able to figure out the different aspects of the seminal work put forward by various researchers in the field of consolidation of virtual machines in cloud computing environment. The state of art has been classified on the basis of distinguished parameters like the underlying architecture, the resources utilized, the consolidation step involved and the objectives achieved. Based upon the survey of the existing literature, the classification of the solutions to the consolidation process has been developed and is presented in Fig. 7. These solutions can be broadly classified into the categories of Optimization and Statistical methods. The optimization methods help in finding the solutions to the problem that tend to maximize or minimize certain defined objectives. In contrast to this, statistical methods work on the raw data and try to analyse it with the help of mathematical formula or technique.

6.1 Optimization Methods

To solve the problem of Dynamic Virtual Machine Consolidation, various optimization techniques have been followed. These optimization techniques can be either heuristic-based or meta-heuristic-based. Heuristics techniques are problem-dependent and are usually based on local search. It may not guarantee an optimal solution but an approximate goal is achieved. These methods can be applied to a large number of problems. Heuristic techniques can be categorized as deterministic and probabilistic. Given a set of input values, the deterministic methods produce a single conclusion whereas, with probabilistic methods, a degree of randomness is involved in the process of solution-finding and objective completion.

The probabilistic methods include game theory models, bin packing techniques, and Markov Chains. Next fit, best fit, worst fit, best fit decreasing are examples of bin packing techniques. However, meta-heuristic techniques help in finding the near-optimal solution and are not problem-dependent. Evolutionary and swarm-based form the categories of metaheuristic methods. The evolutionary methods are inspired by natural mechanisms like reproduction and crossover. There are many evolutionary algorithms proposed by different researchers. A genetic algorithm is one such example. Swarm intelligence-based algorithms are inspired by the behavior of animals or birds in nature. These algorithms are motivated by the interactions and communications occurring between the animals. Algorithms like Ant Colony Optimization, Particle Swarm Optimization, Cuckoo Search, Dragonfly, etc. fall under this category.

6.2 Statistical Methods

Statistical methods of consolidation involve approximation techniques, machine learning techniques, and exact methods. The approximation methods include the methods based on fuzzy logic and queuing theory. Earliest finish time, earliest deadline first, first in first out, Last in first out are examples of queuing model-based methods. Machine learning techniques help in analysing the activity patterns to reduce the human intervention[110]. Regression, k-means, Markov chains, Neural networks, Bayesian networks, reinforcement learning [9, 9], and model predictive control methods come under the category of machine learning techniques while considering the virtual machine consolidation. The exact methods are used to find an exact solution to a small-sized
### Table 11 Meta-heuristics based VM placement techniques

| Authors              | Main idea                                                                 | Objectives                                           | Simulator                               |
|----------------------|----------------------------------------------------------------------------|------------------------------------------------------|-----------------------------------------|
| Gao et al. [93] 2013 | Pareto Set solution for power consumption and resource wastage             | Minimize total power consumption                     | Java Code on Intel Processor            |
| Liu et al. [102] 2014| Incorporation of non-dominated sorting into Genetic Algorithms             | Reduced traffic                                      |                                        |
| Tang and Pan [103] 2015| Use of hybrid Genetic Algorithm Different categories of communication have been considered | Minimized active hosts                               |                                        |
| Ramezani et al. [88] 2016 | Fuzzy logic-based placement policy Triangular membership functions have been employed | Improved resource utilization                        | CloudSim                               |
| Li et al. [20] 2016  | The Bayesian formula has been used to avoid falling into local optima     | Minimized number of migrations                       | CloudSim                               |
| Shabeera et al. [95] 2017 | Defined VM placement as MinDist-VMDataPlacement problem Representation in the form of the weighted graph | Minimized transfer delay                              | MapReduce, CloudSim                    |
| Sonklin et al. [105] 2017 | Reduced problem size with the help of Divide and Conquer Use of a uniform crossover | Reduced computation time                              | Java program / Intel(R) Xeon(R) CPU of 2.40 GHz |
| Liu et al. [96] 2017  | The Pareto front solution to solve different objectives Proposed a pheromone deposition technique | Reduced number of migrations                         |                                        |
| Abdessamia et al. [89] 2017 | Use of Energy consumption as a measure of the fitness function            | Minimized active hosts                               | CloudSim                               |
| Tripathi et al. [90] 2018 | PSO has been modified for VM Placement Multi-objective optimization solved as Pareto set | Reduced power consumption                            | CloudSim                               |
| Yan et al. [91] 2018  | Calculation of host volume using CPU and memory load                       | Better memory utilization                             | OpenStack                              |
| Ashraf and Porres [97] 2018 | Suited for both centralized and distributed architectures Migrations are performed in the neighbourhood only | Maximized number of released machines Minimized number of migrations | Java Code on Intel Core i7-4790 processor |
| More and Ingle [99] 2018 | Load prediction through Support Vector Machine Incorporation of Crow optimization into a dragonfly | Minimized load Reduced energy consumption Lesser migration cost | Java code, Intel I3 processor          |
| Alharbi et al. [98] 2019 | Profile-based approach for VM placement Formulation of problem using constraint satisfaction optimization | Reduced energy consumption                            | Java Code, Intel Core i7-4790 processor |
| Sharma and Saini [106] 2019 | Multi-objective optimization using non-dominated sorting genetic algorithm Randomly generated initial population | Fewer migrations Reduced SLA violations Minimized migrations | CloudSim                               |
| Tripathi et al. [100] 2020 | Use of binary dragonfly Better resource utilization Use of V-shaped transfer function | Optimized resource wastage                            | CloudSim/ Real workload                |
| Medara et al. [109] 2021 | Use of water wave optimization Intended to increase overall resource utilization | Reduced energy consumption                            | WorkflowSim                            |
Table 12 Analysis based on architecture, VM consolidation step, and resources

| Authors                  | Architecture     | VM Consolidation Step | Resources |
|--------------------------|-------------------|-----------------------|-----------|
|                          | Centralized       | Distributed           | HD  | VMS | VMP | CPU | RAM | Disk | BW |
| Beloglazov et al. [19]   | *                 |                       | *   | *   | *   | *   |     |      |     |
| Hongjian Li et al. [20]  | *                 |                       | *   | *   | *   | *   |     |      |     |
| Yahya et al. [21]        | *                 |                       | *   |     | *   | *   |     |      |     |
| Chinni et al. [22]       | *                 |                       | *   |     |     | *   |     |      |     |
| Xiao et al. [23]         | *                 |                       | *   | *   |     | *   |     |      |     |
| Beloglazov et al. [24]   | *                 |                       | *   | *   | *   | *   | *   |      |     |
| Sharma & Saini [26]      | *                 |                       | *   |     |     | *   |     |      |     |
| Farahknkian et al. [27]  | *                 |                       | *   |     |     | *   |     |      |     |
| Minarolli et al. [28]    | *                 |                       | *   |     |     | *   |     |      |     |
| Li et al. [30]           | *                 |                       | *   |     |     | *   |     |      |     |
| Patel et al. [31]        | *                 |                       | *   |     |     | *   |     |      |     |
| Saadi et al. [32]        | *                 |                       | *   |     |     | *   |     |      |     |
| Xie et al. [33]          | *                 |                       | *   | *   | *   | *   |     |      |     |
| Zhou et al. [34]         | *                 |                       | *   |     |     | *   |     |      |     |
| Farahknkian et al. [36]  | *                 |                       | *   |     |     | *   |     |      |     |
| Yadav et al. [37]        | *                 |                       | *   |     | *   | *   |     |      |     |
| Hieu et al. [38]         | *                 |                       | *   |     | *   | *   |     |      |     |
| Abdel et al. [39]        | *                 |                       | *   | *   | *   | *   |     |      |     |
| Yadav et al. [40]        | *                 |                       | *   |     |     | *   |     |      |     |
| Jararweh et al. [41]     | *                 |                       | *   |     |     | *   |     |      |     |
| Yadav et al. [42]        | *                 |                       | *   |     |     | *   |     |      |     |
| Mapetu et al. [43]       | *                 |                       | *   |     | *   | *   |     |      |     |
| Melhem et al. [49]       | *                 |                       | *   |     |     | *   |     |      |     |
| Farahknkian et al. [45]  | *                 |                       | *   |     |     | *   |     |      |     |
| Farahknkian et al. [46]  | *                 |                       | *   |     | *   | *   |     |      |     |
| Zhou et al. [47]         | *                 |                       | *   |     | *   | *   |     |      |     |
| Hopcroft et al. [48]     | *                 |                       | *   |     | *   | *   |     |      |     |
| Li [50]                  | *                 |                       | *   |     | *   | *   |     |      |     |
| Hsieh et al. [51]        | *                 |                       | *   | *   | *   |     |     |      |     |
| Cao [53]                 | *                 |                       | *   |     | *   |     |     |      |     |
| Alboanee et al. [54]     | *                 |                       | *   |     |     | *   |     |      |     |
| Masoumzadeh et al. [55]  | *                 |                       | *   |     |     | *   |     |      |     |
| Rajabzadeh et al. [59]   | *                 |                       | *   |     | *   | *   |     |      |     |
| Mosa et al. [60]         | *                 |                       | *   |     |     | *   |     |      |     |
| Yadav et al. [61]        | *                 |                       | *   |     |     | *   |     |      |     |
| Chang et al. [62]        | *                 |                       | *   |     | *   | *   |     |      |     |
| Wang et al. [63]         | *                 |                       | *   |     | *   |     |     |      |     |
| Li et al. [64]           | *                 |                       | *   |     |     | *   |     |      |     |
| Alaull et al. [57]       | *                 |                       | *   |     |     | *   |     |      |     |
| Shidak et al. [58]       | *                 |                       | *   |     | *   | *   |     |      |     |
| Choudhary et al. [66]    | *                 |                       | *   |     |     | *   |     |      |     |
| Mishra et al. [67]       | *                 |                       | *   |     |     | *   |     |      |     |
| Murtazaev et al. [68]    | *                 |                       | *   |     |     | *   |     |      |     |
| Fu et al. [70]           | *                 |                       | *   |     |     | *   |     |      |     |
| Esfandiarpoor et al. [71]| *                 |                       | *   |     |     | *   |     |      |     |
| Mosa et al. [72]         | *                 |                       | *   |     |     | *   |     |      |     |
| Hammer et al. [73]       | *                 |                       | *   |     |     | *   |     |      |     |
| Wang et al. [63]         | *                 |                       | *   |     |     | *   |     |      |     |
| Chen et al. [75]         | *                 |                       | *   |     |     | *   |     |      |     |
problem. Linear, integer linear programming and constraint satisfaction programming are part of exact methods.

7 Research Gaps

Various gaps in the existing literature have been highlighted in Fig. 8, that have been scrutinized after analyzing the seminal work of various researchers in the relevant field.

(1) With the help of server consolidation, multiple virtual machines are stacked onto a single machine. However, this raises the concern of single-point failure. An occurrence of server down may lead to the destruction of all the virtual machines housed on it.

(2) Certain applications are sensitive to delays such as audio and video conferencing. Resource allocation to such applications requires proper handling and concern.

(3) In the cloud environment, the cloud providers receive simultaneous requests for resources, leading to a race of resources among these requests. In the last decade, researchers have focussed a lot on providing solutions to appropriate and efficient load balancing depending upon current resource utilization. However, predicted future loads and past utilization are not much analysed. Thus, how to balance the VM loads considering the past and future situation is another research challenge.

(4) Shared resources may cause contention during consolidation, further degrading the performance of the applications and raising the issues of SLA violations. Another important challenge that should be taken into account is the interference among VMs, because migrating some of the correlated VMs may degrade service due to network latency especially over WAN links. Therefore, the situation of interference among VMs should also be taken care of while trying to consolidate the VMs.
| Authors                      | Prediction | Objectives | Energy | SLAV | RU | No. of HS | No. of VMM |
|-----------------------------|------------|------------|--------|------|----|----------|-----------|
| Beloglazov et al. [19]      | *          | *          |        |      |    |          |           |
| Hongjian Li et al. [20]     | *          | *          | *      | *   |    |          |           |
| Yahya et al. [21]           | *          | *          | *      |      |    |          |           |
| Chinniaih et al. [22]       | *          |            |        |      |    |          |           |
| Xiao et al. [23]            |            |            | *      |      |    |          |           |
| Beloglazov et al. [24]      |            |            | *      |      |    |          |           |
| Sharma and Saini [26]       | *          |            | *      |      |    |          |           |
| Farahnakian et al. [27]     | *          |            | *      |      |    |          |           |
| Minarolli et al. [28]       | *          | *          |         |      |    |          |           |
| Li et al. [30]              | *          |            |        |      |    |          |           |
| Patel et al. [31]           | *          | *          | *      |      |    |          |           |
| Saadi et al. [32]           | *          | *          | *      |      |    |          |           |
| Xie et al. [33]             | *          |            | *      |      |    |          |           |
| Zhou et al. [34]            |            |            | *      |      |    |          |           |
| Farahnakian et al. [36]     | *          | *          | *      |      |    |          |           |
| Yadav et al. [37]           | *          | *          | *      |      |    |          |           |
| Hieu et al. [38]            | *          | *          | *      |      |    |          |           |
| Abdelsamea et al. [39]      | *          |            | *      |      |    |          |           |
| Yadav et al. [40]           | *          | *          | *      |      |    |          |           |
| Jararweh et al. [41]        | *          | *          | *      |      |    |          |           |
| Yadav et al. [42]           | *          | *          | *      |      |    |          |           |
| Mapetu et al. [43]          | *          | *          | *      |      |    |          |           |
| Melhem et al. [49]          |            | *          | *      |      |    |          |           |
| Farahnakian et al. [45]     | *          | *          | *      |      |    |          |           |
| Farahnakian et al. [46]     | *          | *          | *      |      |    |          |           |
| Zhou et al. [47]            |            | *          | *      |      |    |          |           |
| Hopcroft et al. [48]        | *          | *          | *      |      |    |          |           |
| Li [50]                     | *          |            | *      |      |    |          |           |
| Hsieh et al. [51]           | *          | *          | *      |      |    |          |           |
| Cao [53]                    |            | *          | *      |      |    |          |           |
| Alboaneen et al. [54]       | *          |            | *      |      |    |          |           |
| Masoumzadeh et al. [55]     |            |            | *      | *   |    |          |           |
| Rajabzadeh et al. [59]      | *          | *          | *      |      |    |          |           |
| Mosa et al. [60]            | *          |            | *      |      |    |          |           |
| Yadav et al. [61]           | *          |            | *      |      |    |          |           |
| Chang et al. [62]           | *          |            | *      |      |    |          |           |
| Wang et al. [63]            | *          | *          | *      |      |    |          |           |
| Li et al. [64]              |            |            | *      |      |    |          |           |
| Alaul et al. [57]           | *          | *          | *      |      |    |          |           |
| Shidik et al. [58]          |            | *          | *      |      |    |          |           |
| Choudhary et al. [66]       |            |            | *      |      |    |          |           |
| Mishra et al. [67]          | *          |            | *      |      |    |          |           |
| Murtazaev et al. [68]       | *          |            | *      |      |    |          |           |
| Fu et al. [70]              | *          | *          | *      |      |    |          |           |
| Esfandiarpoor et al. [71]   |            |            | *      |      |    |          |           |
| Hammer et al. [73]          | *          | *          | *      |      |    |          |           |
| Wang et al. [63]            |            |            | *      |      |    |          |           |
| Chen et al. [75]            | *          | *          | *      |      |    |          |           |
| Arani [76]                  | *          |            | *      |      |    |          |           |
Researchers have paid a lot of interest to the single VM migration. But the improvement in the system’s performance demands the migration of multiple Virtual machines. Thus, scheduling multiple VM migrations is the main challenge to optimize the performance of the cloud data center.

Current load balancing and load detection techniques are static in nature and may not be suitable for dynamic workloads. The dynamic nature of the workload demands the development of adaptive schemes for load detection.

Maximizing the RAM utilization capacity may increase the chances of new user requests being dropped. Also, higher levels of CPU utilization raise the server temperature. This, in turn, demands cooling systems leading to more energy consumption. To guarantee the important goals of energy efficiency and QoS, the mapping of VMs to PMs should be intended towards maintaining a trade-off between the efficient utilization of resources and minimized SLA violations.

Resource management is the biggest challenge of cloud computing since cloud environment has multiple resources types. Whenever the cloud service provider receives a user request, the resources are allocated according to the need of the users. Where CPU is the most used resource, graphics processing unit (GPU) has also been incorporated to cloud computing environment for handling graphics-intensive workloads. According to the studies, integration of both the PUs can accelerate the performance of the system.

VMP can be considered as an optimization problem of the consolidation process which is inherently discrete in nature. The existing optimization techniques often fall in local optima thereby, not being able to reach the global best solution. Thus, designing new metaheuristic algorithms or improving the existing techniques to overcome this issue of the Virtual Machine Placement problem is an important area of research.

**8 Proposed VM Consolidation Framework**

In an attempt to fulfil some of the gaps of the existing literature, we propose a VMC as shown in Fig. 9. While selecting the overloaded host, emphasis should be laid on considering both current and past/future utilizations. This has led us to focus on the scaled estimation techniques for host overload detection. A scale estimator helps in forecasting the future

---

**Table 13 (continued)**

| Authors            | Prediction | Objectives |
|--------------------|------------|------------|
| Refs.              | Predictive | Non-Predictive | Energy | SLA | RU | No. of HS | No. of VMM |
| Gupta et al. [77]  | *          | *          | *      |     |     |            |            |
| Haghshenas et al. [79] | *          | *          |        |     |     |            |            |
| Gao et al. [93]    | *          | *          |        |     |     |            |            |
| Dupont et al. [80] | *          | *          |        |     |     |            |            |
| Zhang et al. [81]  | *          | *          |        |     |     |            |            |
| Tchana et al. [82] | *          | *          |        |     |     |            |            |
| Huang et al. [86]  | *          | *          |        |     |     |            |            |
| Ramezani et al. [88] | *          | *          |        |     |     |            |            |
| Abdessamia et al. [89] | *          | *          |        |     |     |            |            |
| Tripathi et al. [90] | *          | *          |        |     |     |            |            |
| Yan et al. [91]    | *          | *          |        |     |     |            |            |
| Liu et al. [96]    | *          | *          |        |     |     |            |            |
| Ashraf et al. [97] | *          | *          |        |     |     |            |            |
| Alharbi et al. [98] | *          | *          |        |     |     |            |            |
| More et al. [99]   | *          | *          |        |     |     |            |            |
| Tripathi et al. [100] | *          | *          |        |     |     |            |            |
| Liu et al. [102]   | *          | *          |        |     |     |            |            |
| Tang et al. [103]  | *          | *          |        |     |     |            |            |
| Kaaouache et al. [104] | *          | *          |        |     |     |            |            |
| Sonklin et al. [105] | *          | *          |        |     |     |            |            |
| Sharma et al. [106] | *          | *          |        |     |     |            |            |
| Zang et al. [9]    | *          | *          |        |     |     |            |            |

*RU Resource Utilization, HS Host Shutdown, VMM Virtual Machine Migration
*Yes
values depending upon the current and past set of certain values. Bi-weight mid correlation shows it advantages over standard Pearson correlation techniques in terms of robustness and resistance of efficiency. Which helps in dealing with one of the major challenges of VM selection in the process of VM Consolidation.

Being NP hard in nature, the problem of VM placement can be best resolved with the help of the metaheuristic techniques. Since, Dragonfly has proved to be superior in terms of performance as compared best known meta-heuristic techniques [112], we propose use an adaptive binary dragonfly method for solving the problem of VM placement.

9 Conclusion

The advances in virtualization technology have proved to be a boon for revolutionizing IT and academics. The revolution in the IT industry over the past decade has led to an
increased demand of cloud computing resources and has resulted into a great pace escalation in the number of users requesting for resources in Cloud Data Centers. This results in a situation where the tasks compete for resources and the outcome is huge energy consumption. Such a condition demands for appropriate resource management mechanisms and energy efficient schemes. Within a data center, the dynamic consolidation process is considered to be one of the most efficient methods for reducing the energy consumption. Virtualization and consolidation of VMs have been widely accepted by researchers for improving resource utilization in the CDCs. Various static and dynamic approaches used by different researchers for the Virtual Machine Consolidation Process have been discussed in this research paper. In the current research, a detailed analysis of consolidation time selection, VM selection, and VM deployment has been carried out. Observing the gaps in the literature, a framework for VM Consolidation has also been proposed, the implementation of the same will be carried out in future.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Buyya R, Broberg J, Goschinski A (2011) Market-oriented cloud computing and the cloudbus toolkit. Cloud Comput. https://doi.org/10.1002/9780470940105
2. Jain N, Choudhary S (2016) Overview of virtualization in cloud computing. 2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016, doi: https://doi.org/10.1109/CDAN.2016.7570950.
3. Dayarathna M, Wen Y, Fan R (2016) Data center energy consumption modeling: a survey. IEEE Commun Surv Tutor 18(1):732–794. https://doi.org/10.1109/COMST.2015.2481183
4. Avergerinou M, Bertoldi P, Castellazzi L (2017) Trends in data centre energy consumption under the European code of conduct for data centre energy efficiency. Energies (Basel). https://doi.org/10.3390/en10101470
5. Kaur S, Kumar Y, Kumar S (2021) Soft computing techniques for energy consumption and resource aware allocation on cloud: a progress and systematic review. Adv Soft Comput Techn Data Sci IoT Cloud Comput Stud Big Data 89:191–213
6. Daid R, Kumar Y, Hu YC, Chen WL (2021) An effective scheduling in data centres for efficient CPU usage and service level agreement fulfilment using machine learning. Conn Sci 33(4):954–974. https://doi.org/10.1080/095940091.2021.1926929
7. Fan X, Weber W, Barroso LA (2007) Power provisioning for a warehouse-sized computer. ACM Sigarch Comput Architect News 35(2):13–23. https://doi.org/10.1145/1273440.1250665
8. “How much energy do data centers consume?” https://www.techtarget.com/searchdatacenter/tp/How-much-energy-do-data-centers-consume. Accessed 23 Sep 2022
9. Zeng J, Ding D, Kang K, Xie HM, Yin Q (2022) Adaptive DRL-based virtual machine consolidation in energy-efficient cloud
10. Kumar Y, Kaul S, Sood K Effective use of the Machine learning approaches on different clouds. Available: https://ssrn.com/abstract=3355203
11. Donyangard N, Mostafa V, Arani G (2019) Multiobjective virtual machine placement mechanisms using nature-inspired metaheuristic algorithms in cloud environments: a comprehensive review. Int J Commun Syst 5:1–32. https://doi.org/10.1002/da.4068
12. Zolfaghari R, Rahmani AM (2020) Virtual machine consolidation in cloud computing systems: challenges and future trends. Wirel Personal Commun. https://doi.org/10.1007/s11277-020-07682-8
13. Bermejo B, Juiz C (2020) Virtual machine consolidation: a systematic review of its overhead influencing factors. J Supercomput. https://doi.org/10.1007/s11227-019-03025-y
14. Pourgehebleh B, Anvigh AA, Ramtin AR, Mohammad B (2021) The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments. Clust Comput 24(3):2673–2696. https://doi.org/10.1007/s10586-021-03294-4
15. Ashraf A, Byholm B, Porres I (2016) Distributed virtual machine consolidation: a systematic mapping study. TUCS Technical Report no. 2017
16. Khan A, Paplinski A, Khan AM, Murshed M, Buyya R (2018) Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: a review. Sustain Cloud Energy Serv. https://doi.org/10.1007/978-3-319-62238-5
17. Singh BP, Kumar SA, Gao XZ, Kohli M, Katiyar S (2020) A study on energy consumption of DVFS and simple VM consolidation policies in cloud computing data centers using CloudSim Toolkit. Wirel Pers Commun 112(2):729–741. https://doi.org/10.1007/s11277-020-07070-2
18. Masdari M, Nabavi SS, Ahmadi V (2016) Journal of network and computer applications an overview of virtual machine placement schemes in cloud computing. J Netw Comput Appl 66:106–127. https://doi.org/10.1016/j.jnca.2016.01.011
19. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for Cloud Computing. Futur Gener Comput Syst 28(5):755–768. https://doi.org/10.1016/j.future.2011.04.017
20. Li H, Zhu G, Cui C, Tang H, Dou Y, He C (2016) Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing. Computing 98(3):303–317. https://doi.org/10.1007/s10077-015-0467-4
21. Yahya S, Fard Z, Adabi S, Ahmadi MR (2017) A dynamic VM consolidation technique for QoS and energy consumption in cloud environment. J Supercomput 73(10):4347–4368. https://doi.org/10.1007/s11227-017-1920-6
22. Chinnaiah V, Gudi Pudi S, Somasundaram TS, Basha SS (2018) A cloud resource allocation strategy based on fitness based live migration and clustering. Wirel Pers Commun 98(3):2943–2958. https://doi.org/10.1007/s11227-017-5009-2
23. Xiao H, Hu Z, Li K (2019) Multi-objective VM consolidation based on thresholds and ant colony system in cloud computing. IEEE Access 7:53441–53453. https://doi.org/10.1109/ACCESS.2019.2912722
24. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurr Comput Pract Exp 24(13):1397–1420. https://doi.org/10.1002/cpe.1867
25. Chang YC, Chang RS, Chuang FW (2014) A predictive method for workload forecasting in the cloud environment. Lect Notes Electr Eng 260:577–585. https://doi.org/10.1007/978-94-007-7262-5_65

Adaptive Computational Solutions to Energy Efficiency in Cloud Computing Environment Using...
26. Sharma O, Saini H (2016) VM consolidation for cloud data Center using median based threshold approach. Procedia 89:27–33. https://doi.org/10.1016/j.procs.2016.06.005
27. Farahnakian F, Bahsoon R, Liljeberg P, Pahikkala T (2017) Self-adaptive resource management system in IaaS clouds. IEEE Int Conf Cloud Comput CLOUD. https://doi.org/10.1109/CLOUD.2016.77
28. Minarolli D, Mazreka A, Freisleben B (2017) Tackling uncertainty in long-term predictions for host overload and underload detection in cloud computing. J Cloud Comput. https://doi.org/10.1186/s13677-017-0074-3
29. Dambreville A, Tomašik J, Cohen J, Dufoulon F (2017) Load prediction for energy-aware scheduling for cloud computing platforms. In: Proceedings - international conference on distributed computing systems, pp 2604–2607. doi: https://doi.org/10.1109/ICDCS.2017.201
30. Li Z, Yan C, Yu X, Yu N (2017) Bayesian network-based Virtual Machines consolidation method. Futur Gener Comput Syst 69:75–87. https://doi.org/10.1016/j.future.2016.12.008
31. Patel N, Patel H (2017) Energy efficient strategy for placement of virtual machines selected from underloaded servers in compute cloud. J King Saud Univ. https://doi.org/10.1016/j.jsuci.2017.11.003
32. Saadi Y, El Kafhali S (2020) Energy-efficient strategy for virtual machine consolidation in cloud environment. Soft Comput 24(19):14845–14859. https://doi.org/10.1007/s00500-020-04839-2
33. Xie L, Chen S, Shen W, Miao H (2018) A novel self-adaptive VM consolidation strategy using dynamic multi-thresholds in IaaS clouds. Future Internet 10(6):1–18. https://doi.org/10.3390/ fi10060052
34. Zhou H, Li Q, Choo KKR, Zhu H (2018) DADTA: A novel adaptive strategy for energy and performance efficient virtual machine consolidation. J Parall Distrib Comput 121:15–26. https://doi.org/10.1016/j.jpdc.2018.06.011
35. Karim ME, Maswood MMS, Das S, Alharbi AG (2021) BHyPreC: a novel Bi-LSTM based hybrid recurrent neural network model to predict the CPU workload of cloud virtual machine. IEEE Access 9:131476–131495. https://doi.org/10.1109/ACCESS.2021.3113714
36. Farahnakian F, Liljeberg P, Plosila J (2013) LiRCUP: linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. Proceedings—39th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2013, pp. 357–364, doi: https://doi.org/10.1109/SEAA.2013.23
37. Yadav R, Zhang W, Kawaiyta O, Singh PR, Elgendy IA, Tian YC (2018) Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing. IEEE Access 6(9):55923–55936. https://doi.org/10.1109/ACCESS.2018.2872750
38. Hsieh SY, Liu CS, Buyya R, Zomaya AY (2020) Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers using utilization prediction model. IEEE Trans Cloud Comput 7(2):524–536. https://doi.org/10.1109/TCCT.2016.2617374
39. Minhaj Ahmad Khan (2021) An efficient energy-aware approach for dynamic VM consolidation on cloud platforms. Cluster Comput 24:3293–3310. https://doi.org/10.1007/s10586-021-03341-0
40. Farahnakian F, Pahikkala T, Liljeberg P, Plosila J (2013) Energy aware consolidation algorithm based on K-nearest neighbor regression for cloud data centers. In: Proceedings—2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC 2013, June 2014, pp 256–259, doi: https://doi.org/10.1109/UCC.2013.51
41. Mapetu JPB, Kong L, Chen Z (2020) A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing. J Supercomput. https://doi.org/10.1007/s11227-020-03494-6
42. Hieu NT, Di Francesco M, Yla-Jaaski A (2017) Virtual machine consolidation strategy using dynamic multi-thresholds in cloud computing. J Cloud Comput. https://doi.org/10.1186/s13677-016-0059-7
58. Shidik GF, Mustofa AK (2016) Improvement of energy efficiency at cloud data center based on fuzzy Markov Normal Algorithm VM selection in dynamic VM consolidation. Int Rev Comput Softw 11(6):511–520. https://doi.org/10.15866/irecos.v11i6.9473
59. Rajabzadeh M, Toroghi A (2017) Energy-aware framework with Markov chain-based parallel simulated annealing algorithm for dynamic management of virtual machines in cloud data centers. J Supercomput 73(5):2001–2017. https://doi.org/10.1007/s11227-016-1900-y
60. Mosa A, Sakellariou R (2017) Virtual machine consolidation for cloud data centers using parameter-based adaptive allocation, doi: https://doi.org/10.1145/3123779.3123807
61. Yadav R, Zhang W, Chen H, Guo T (2017) MuMs: energy-aware VM selection scheme for cloud data center. In: Proceedings—international workshop on database and expert systems applications, DEXA, pp 132–136. doi: https://doi.org/10.1109/DEXA.2017.43.
62. Chang Y, Gu C, Luo F, Fan G, Fu W (2018) Energy efficient resource selection and allocation strategy for virtual machine consolidation in cloud datacenters. IEICE Trans Inf Syst E101D(7):1816–1827. https://doi.org/10.1587/transinf.2017EDP7321
63. Wang H, Tianfield H (2018) Energy-aware dynamic virtual machine consolidation for cloud datacenters. IEEE Access 6:15259–15273. https://doi.org/10.1109/ACCESS.2018.2813541
64. Li H, Li W, Wang H, Wang J (2018) An optimization of virtual machine selection and placement by using memory content similarity for server consolidation in cloud. Futur Gener Comput Syst 84:98–107. https://doi.org/10.1016/j.future.2018.02.026
65. Lin YL, Tao F, Wang F, Zhang L (2018) An iterative budget algorithm for dynamic virtual machine consolidation under cloud computing environment. IEEE Trans Serv Comput 14(1):30–43. https://doi.org/10.1109/TSC.2018.2793209
66. Choudhary A, Govil MC, Singh G, Awasthi LK, Pilli ES (2019) Energy-efficient fuzzy-based approach for dynamic virtual machine consolidation. Int J Grid Util Comput 10(4):308–325. https://doi.org/10.1504/IJGUC.2019.100863
67. Mishra M, Sahoo A (2011) On theory of vm placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach. In: Proceedings—2011 IEEE 4th international conference on cloud computing, CLOUD 2011, pp. 275–282. doi: https://doi.org/10.1109/CLOUD.2011.38.
68. Murtazaev A, Oh S (2011) Sercon: server consolidation algorithm using live migration of virtual machines for green computing. IETE Tech Rev 28(3):2011
69. Huang Z, Tsang DHK (2012) SLA guaranteed virtual machine consolidation for computing clouds. IEEE Int Conf Commun. https://doi.org/10.1109/ICC.2012.6363970
70. Fu X, Zhou C (2015) Virtual machine selection and placement for dynamic consolidation in Cloud computing environment. Front Comput Sci 9(2):322–330. https://doi.org/10.1007/s11704-015-4286-8
71. Esfandiarpoor S, Pahlavan A, Goudarzi M (2015) Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing. Comput Electr Eng 42:74–89. https://doi.org/10.1016/j.compeleceng.2014.09.005
72. Mosa A, Paton NW (2016) “Optimizing virtual machine placement for energy and SLA in clouds using utility functions. J Cloud Comput. https://doi.org/10.1186/s13677-016-0067-7
73. Hammer HL, Yazidi A, Begnum K (2017) An inhomogeneous hidden markov model for efficient virtual machine placement in cloud computing environments. J Forecast 420(407–420):2017. https://doi.org/10.1002/for.2441
74. Zahedifar SY, Ahmadi MR, Adabi S (2017) A dynamic VM consolidation technique for QoS and energy consumption in cloud environment. J Supercomput 73(10):4347–4368. https://doi.org/10.1007/s11227-017-1908-8
75. Chen T, Zhu Y, Gao X, Kong L, Chen G, Wang Y (2018) Improving resource utilization via virtual machine placement in data center networks. Mobile Netw Appl 23(2):227–238. https://doi.org/10.1007/s11227-017-0925-7
76. Ghobaei-Arani M, Rahmani AA, Shamsi M, Rasouli-Kenari A (2018) A learning-based approach for virtual machine placement in cloud data centers. Int J Commun Syst 31(8):1–18. https://doi.org/10.1002/dac.3537
77. Gupta MK, Amgoth T (2018) Resource-aware virtual machine placement algorithm for IaaS cloud. J Supercomput 74(1):122–140. https://doi.org/10.1007/s11227-017-2112-9
78. Mishra SK et al (2018) Energy-efficient VM-placement in cloud data center. Sustain Comput 20:48–55. https://doi.org/10.1016/j.suscom.2018.01.002
79. Haghshenas K, Mohammadi S (2020) Prediction-based underutilized and destination host selection approaches for energy-efficient dynamic VM consolidation in data centers. J Supercomput 76(12):10240–10257. https://doi.org/10.1007/s11227-020-03248-4
80. Dupont C, Giuliani G, Hermenier F, Schulze T, Somov A (2012) An energy aware framework for virtual machine placement in cloud federated data centres. In: Proceedings of the 3rd international conference on future energy systems: “Where Energy, Computing and Communication Meet”, e-Energy 2012, doi: https://doi.org/10.1145/2208828.2208832
81. Zhang L, Zhuang Y, Zhu W (2013) Constraint programming based virtual cloud resources allocation model. Int J Hybrid Inform Technol 6(6):333–344. https://doi.org/10.14257/ijhit.2013.6.6.30
82. Tchana A, De Palma N, Saffieddine I, Hagimont D (2012) Software consolidation as an efficient energy and cost saving solution. Futur Gener Comput Syst 58:1–12. https://doi.org/10.1016/j.future.2015.11.027
83. “FIT4Green EU Project.” http://www.fit4green.eu
84. Tseng FH, Chen CY, Der Chou L, Chao HC, Niu JW (2015) Service-oriented virtual machine placement optimization for green data center. Mobile Netw Appl 20(5):556–566. https://doi.org/10.1007/s11036-015-0600-9
85. Zeng D, Guo S, Huang H, Yu S, Leung VCM (2015) Optimal VM placement in data centres with architectural and resource constraints. Int J Auton Adapt Commun Syst 8(4):392–406. https://doi.org/10.1007/s11036-015-07318-7
86. Huang Z, Tsang DHK (2016) M-convex VM consolidation: towards a better VM workload consolidation. IEEE Trans Cloud Comput 4(4):415–428. https://doi.org/10.1109/TCC.2014.2369423
87. Gurobi optimization Reference Manual (2013). http://www.gurobi.com
88. Ramezani F, Naderpour M, Lu J (2016) A multi-objective optimization model for virtual machine mapping in cloud data centres. In: 2016 IEEE international conference on fuzzy systems, FUZZ-IEEE 2016, pp. 1259–1265. doi: https://doi.org/10.1109/FUZZ-IEEE.2016.7737833.
89. Abdessamia F, Tai Y, Zhang WZ, Shafiq M (2017) An improved particle swarm optimization for energy-efficiency virtual machine placement. In: Proceedings—5th international conference on cloud computing research and innovation, ICCCRI 2017, pp. 7–13. doi: https://doi.org/10.1109/ICCCRI.2017.9.
90. Tripathi A, Pathak I, Vidyarthi DP (2018) Energy efficient VM placement for effective resource utilization using modified binary PSO. Comput J 61(6):832–846. https://doi.org/10.1093/comjnl/bxx096
91. Yan J, Zhang H, Xu H, Zhang Z (2018) Discrete PSO-based workload optimization in virtual machine placement. Pers
103. Tang M, Pan S (2015) A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers.

Neural Process Lett 41(2):211–221. https://doi.org/10.1007/s11063-014-9339-8

104. Kaouache MA, Bouamama S (2015) Solving bin packing problem with a hybrid genetic algorithm for VM placement in cloud. Procedia Comput Sci 60(1):1061–1069. https://doi.org/10.1016/j.procs.2015.08.151

105. Sonklin C, Tang M, Tian CY (2017) A decrease-and-conquer genetic algorithm for energy efficient virtual machine placement in data centers. In: Proceedings—2017 IEEE 15th international conference on industrial informatics, INDIN 2017, pp. 135–140. doi: https://doi.org/10.1109/INDIN.2017.8104760.

106. O. Sharma and H. Saini, “Energy and SLA Efficient Virtual Machine Placement in Cloud Environment Using Non-Dominated Sorting Genetic Algorithm,” vol. 13, no. 1, pp. 1–16, 2019, doi: https://doi.org/10.4018/IJISP.2019010101.

107. H. Teyeb, A. Balma, S. Tata, and N. Ben Hadj-Alouan, “Traffic-aware virtual machine migration scheduling problem in geographically distributed data centers,” IEEE International Conference on Cloud Computing, CLOUD, pp. 798–801, 2017, doi: https://doi.org/10.1109/CLOUD.2016.108.

108. Li R, Zheng Q, Li X, Yan Z (2017) Multi-objective optimization for rebalancing virtual machine placement. Futur Gener Comput Syst. https://doi.org/10.1016/j.future.2017.08.027

109. Medara R, Singh RS (2021) and Amit, “Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization.” Simul Model Pract Theory 110:102323. https://doi.org/10.1016/J.SIMPAT.2021.102323

110. Kumar Y, Kaul S, Hu YC (2022) Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey. Sustain Comput 36:100780. https://doi.org/10.1016/J.SUSCOM.2022.100780

111. Rachael S, Enda H, Enda B (2022) Applying Reinforcement Learning towards automating energy efficient virtual machine consolidation in cloud data centers. Inf Syst 107:25

112. Amini Z, Maen M, Jahangir MR (2018) Providing a load balancing method based on dragonfly optimization algorithm for resource allocation in cloud computing. Int J Netw Distrib Comput 6(1):35–42. https://doi.org/10.2991/ijndc.2018.6.1.4

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.