IJICTR
International Journal of Information & Communication Technology Research

Volume 14- Number 2 – 2022 (14 -22)

Cost Reduction Using SLA-Aware Genetic Algorithm for Consolidation of Virtual Machines in Cloud Data Centers

Hossein Monshizadeh Naeen*
Department of Computer Engineering,
Neyshabur Branch, Islamic Azad University, Neyshabur, Iran
monshizadeh@iau-neyshabur.ac.ir

Received: 12 April 2022 – Revised: 23 May 2022 - Accepted: 20 June 2022

Abstract—**Cloud computing is a computing model which uses network facilities to provision, use and deliver computing services. Nowadays, the issue of reducing energy consumption has become very important alongside the efficiency for Cloud service providers. Dynamic virtual machine (VM) consolidation is a technology that has been used for energy efficient computing in Cloud data centers. In this paper, we offer solutions to reduce overall costs, including energy consumption and service level agreement (SLA) violation. To consolidate VMs into a smaller number of physical machines, a novel SLA-aware VM placement method based on genetic algorithms is presented. In order to make the VM placement algorithm be SLA-aware, the proposed approach considers workloads as non-stationary stochastic processes, and automatically approximates them as stationary processes using a novel dynamic sliding window algorithm. Simulation results in the CloudSim toolkit confirms that the proposed virtual server consolidation algorithms in this paper provides significant total cost savings (evaluated by ESV metric), which is about 45% better than the best of the benchmark algorithms.**

Keywords: component; Cloud Computing; Green IT; SLA violation; VM Consolidation; Genetic Algorithms.

Article type: Research Article

Publisher: ICT Research Institute

[DOI: 10.52547/itrc.14.2.14]

I. INTRODUCTION

In recent years, Cloud computing offers utility-oriented IT services, based on a pay-as-you-go model to worldwide users. It is not completely a new concept, however, current observations suggest a lucrative market for investing in it. The widespread presence of large companies such as Sun Microsystems, Amazon, Google, Microsoft, etc. in the competitive field of cloud computing shows the rapid development and dominance of this processing model in the world of information technology [1]. Cloud based services provide on-demand access to shared resources, enabling companies to outsource their IT infrastructures, and Cloud providers supply virtualized resources to handle the ever-increasing demands of Cloud users. As a result, Cloud data centers consume a significant amount of energy in order to supply services

* Corresponding Author

to a wide range of users, which increases operating costs and CO_2 emissions [2].

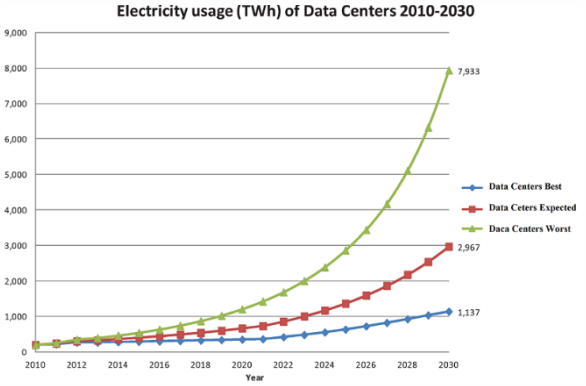

Fig. 1. Global electricity demand of data centers 2010–2030 [3]

Andrea et al. [3] have discussed about the trends of power consumption by data centers (shown in Fig. 1) considering three scenarios, in which they insist on the importance of power management in data centers. The worst-case scenario is exorbitant, however not totally unrealistic. To meet user expectations in a cost-effective manner, Cloud service providers should minimize energy consumption while considering service level agreements (SLAs) [2]. SLAs define the quality of service (QoS) guarantees which are stated in the contracts between Cloud service providers and their customers. For instance, an SLA may determine that the response time of a request must be in a certain duration (e.g. 200ms) and the penalty could be that if the service provider violate this agreement, the fee paid to the service provider would be reduced for a limited time. Therefore, there is an obvious trade-off between energy consumption and QoS establishment.

Nowadays, virtual machine (VM) consolidation approaches are employed in Cloud datacenters to decrease energy consumption by placing VMs on a reduced number of physical hosts [4]. To satisfy the QoS, VM consolidation approaches use live VM migration to transfer a VM from an overloaded host to another. In a live VM migration (also called relocation or real-time migration[5]) the VM state (memory pages and processor state) are transparently transferred from one physical machine (PM) to another while the VM is in use [6]. There are various approaches to transfer VM state from one PM to another which are explained in [7]. However, a VM migration leads to SLA violations: (i) Performance degradation of the applications running on the migrating VM during the relocation process [8]. (ii) A short downtime happens when at the final phase of the migration process [9]. This effect is shown if Fig. 2. Simple consolidation policies may lead to many live migrations.

Workloads on VMs are dynamic and the variation of workloads on VMs and PMs makes the problem more challenging (i.e., overload and underload conditions may happen). Based on the things discussed above, in a Cloud environment with heterogeneous physical hosts and virtual machines, the consolidation problem can be defined as: to determine what time, which VMs, and where should be migrated to minimized the total cost in the data center. Therefore, there is a need for efficient methods for VM consolidation to establish an equilibrium between energy consumption and SLA violation.

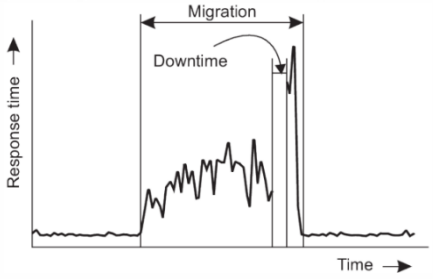

Fig. 2. The effect on the response time of a service while migrating its underlying virtual machine [9].

As mentioned, reducing energy consumption by consolidation techniques increases the likelihood of SLA violations. Therefore, the establishment of SLA has been considered in various studies by considering overload detection and/or overload prediction of executing physical servers [10-13]. Some other works such as [14] and [15] consider overload probability before VM placement to reduce SLA violations. There are few works, such as the works done by Naeen et al. [4, 15] and Beloglazov et al. [16], that have mentioned the issue of non-stationary nature of workload data. However, [4] and [16] solutions are provided for workloads with Markov property, and the solution presented in [15] is not workload adaptive and uses a static sliding window method to deal with non-stationary nature of the workloads.

In this work a genetic algorithm (GA) based approach for SLA-aware VM placement is introduced, which dynamically adapts itself to the non-stationary workload data variations in order to reduce the total costs. The main costs considered in this paper are energy consumption and SLA violation. The most important contributions of this work include:

- Proposing a cost efficient (i.e. energy efficient and SLA-aware) VM consolidation approach for Cloud data centers with non-stationary workloads using a new GA-based VM placement algorithm; the proposed approach reduces the total cost in the system which is evaluated by ESV metric (see section IV) and also the total number of migrations in data centers.

- Presenting an algorithm that dynamically adapts the sliding window lengths to workload variations. The GA-based placement algorithm uses this dynamic size sliding window method to perform SLA-aware VM placement.

- The performance of the proposed algorithms is evaluated by the means of extensive simulations using CloudSim Toolkit with real workload data.

Our proposed This paper continues by reviewing background of VM consolidation in section II. Section III discusses the proposed system model and main SLA violation factors in this study. Section IV describes the experimental results; evaluation metrics and analysis of results are also presented in this section. Finally, the conclusion and future possible directions are discussed in section V.

[DOI: 10.52547/itrc.14.2.14]

II. Related Works

In virtualized data center, most running virtual hosts only operate on a small part of the total available resources [17]. Two sample VM's CPU utilization are shown in Fig. 3. Thus, multiple underutilized servers may take up more space and allocate more resources than can be justified by their workloads. This problem is called server sprawl. Virtualization and live migration can be used to dynamically consolidate to a limited number of PMs and switch idle ones off [18]. Increasing resource utilization and, thus, reducing the number of active PMs in data centers have considerable advantages such as saving energy and other costs [19]. The VM consolidation problem can be divided into several sub-problems (such as VM placement, overload and underload detection, and VM selection) and each can be considered independently. When the requested resources on a PM are more than its capacity, the PM is known as overloaded. This condition leads to performance degradation of the residing VMs on the overloaded PM (i.e., SLA is violated). Thus, it's necessary to avoid this condition. On the other hand, if all the VMs located on a PM can be migrated out of the PM without the occurrence of a new overload condition, the host is considered to be underloaded. Underload detection is important, since low loaded PMs can turn to low-power mode or switch off. VM placement algorithms are employed to determine the new hosts of the migrating VMs which are selected for migration from overloaded PMs (via VM selection algorithms).

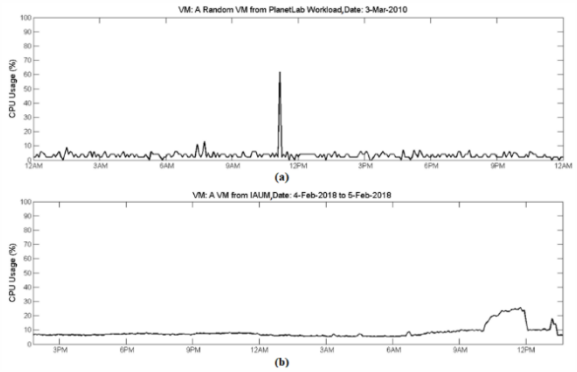

Fig. 3. (a) A random VM selected from PlanetLab - (b) A random VM Selected from IAUM Data Center

However, not all studies have necessarily followed such a division; nevertheless, generally, they try to manage resources in order to consider a trade-off between energy consumption and other performance criteria (such as, number of migrations [15, 20], SLA violations [21], maintenance and reliability [22], etc.) in Cloud systems. VM consolidation algorithms are known as NP-hard algorithms [23], and the implementation methods can be divided into two main categories: exact methods and approximate methods. Exact methods are applicable only on small size inputs (Clouds with small number of PMs and VMs).

One of the earliest works on VM consolidation in large Cloud data centers has been done by Nathuji and Schwan [24]. They have split the resource management problem into two levels. With the assumption that VMs have a power-aware OS, at one level, a local manager cooperates in power management on its host. At a global level, a manager finds the VM migration map. To consolidate the VMs, they have carry out the VM to PM mapping process periodically without performing any overload or underload detection. Verma et al. [25] have performed dynamic VM consolidation for heterogeneous environments, considering it as a continuous optimization problem. In their suggested solution, VM placement is optimized and efficiency increases by considering both the migration cost and the energy cost in every time period. Like [26], they have presented a greedy approach to solve the VM placement sub-problem considering it as a various sized and cost bin-packing problem; Both articles do not consider SLAs and thus, quality of services may decrease due to workload variations.

A known study that finds the solution for dynamic VM consolidation problem is [2]. The authors have proposed a placement algorithm namely Modified Best Fit Decreasing (MBFD) for a virtualized datacenter with heterogeneous hosts in which a VM is greedily allocated to a host with the least increase in energy consumption. The authors have investigated a couple of VM selection policies named Minimization of Migration (MM), Highest Potential Growth (HPG), and Random Choice (RC) policy. The MM policy chooses the least number of virtual machines needed to relocate from a PM to lower the CPU utilization below a predefined upper utilization threshold if a PM is overloaded. In case of an overload condition, the HPG policy migrates VMS that have the lowest CPU utilization relative to the CPU capacity defined by the VM parameters, with the goal to minimized the potential increase of PM's utilization and prevent future SLA violation. The aforementioned work was extended by presenting various heuristic algorithms in [8]. In the study [8] the authors have presented a two new VM selection policies named Minimum Migration Time (MMT) and Maximum Correlation (MC) policy. In MMT policy, a VM that requires the minimum time to complete a relocation relative to other VMs on the same overloaded PM. Migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available to the destination PM. The MC policy chooses a VM that has the maximum correlation of resource utilization with other VMs. Simulation results indicate that MMT outperforms other VM selection policies in case of SLA violation due to migration. Hence, MMT is one of the most popular policies that have been used in the next works by researchers.

Arianyan et al. in [27] have worked on overloading host detection and VM selection as efficient approaches to reduce energy consumption in cloud data centers. The authors have presented fuzzy based solutions for the whole phases of server consolidation in another work [11]. In [11] they have presented a method in which different criteria such as residual resources, potential, bandwidth, RAM capacity and power consumption in servers are considered for selecting the destination of a virtual machine. According to their model, hosts are scored by using a fuzzy system based on the mentioned criteria and the machine with the best score is selected as the new place of a virtual machine. A sequential optimization based solution for power management is proposed by Kusic et al. [28] which is

[DOI: 10.52547/itrc.14.2.14]

solved by a limited look-ahead control. They have minimized both power consumption and SLA violations. They use Kalman filter [29] to predict overloaded hosts. It is claimed in [30] that their optimization process takes up to 30 minutes for 15 hosts, which means it's not a scalable solution.

The authors in [31] have proposed a VM placement algorithm using queuing theory for workloads with burstiness pattern. They have modeled the resource utilization of VMs as a two state Markov chain to represent burstiness. Their system is based on the assumption that the input workloads are stationary, known a priori, and have burstiness pattern. A Markov chain is used to predict next state of a server, and in case of an overload prediction on a PM, some VMs should be migrated out of the PM. In a work done by other authors [4] the workloads on the PMs are considered as Markova processes. However, with a similar idea to [31] they form a Markov chain on each PM based on the resource utilization (without burstiness assumption) to predict future state of the PMs. They have presented three overload detection policies named Deferred Overload Detection (DOD), Immediate Overload Detection (IOD), and Prediction-based Overload Detection (POD). According to DOD, the overload detection is deferred to the time that an actual overload happens. In POD, if the next state of the host is predicted to be overloaded, the PM is considered as overloaded and some VMs should migrate out before a real overload happens. They use a long-term prediction method in IOD policy, to move VMs out of a PM. Results show that IOD has better results for SLA than POD and DOD, while it leads to slightly more energy consumption in the system. Naeen et al. [15] have presented a heuristic VM placement algorithm named stochastic process-based BFD (SBBFD), which considers the workloads on PMs as stochastic processes. SBBFD reduces energy consumption, number of migrations, and SLA violation, but works based on a single static size sliding window method and it is not self-adaptive to non-stationary workloads.

However, meta-heuristic algorithms are more effective for finding optimal solutions[32]. Farahnakian et al. [33] proposed an ant colony optimization (ACO) system for energy efficient consolidation of VMs. They use a K-nearest neighbor (KNN) method to predict overload conditions to prevent SLA violations. A meta-heuristic approach named modified particle swarm optimization is proposed by Li et al. [34] to reduce energy consumption and QoS optimization. The Authors in [35] and [36] have proposed GA based placement algorithms for energy efficient VM placement in Cloud data centers; the methods presented in these two studies cannot consider the SLAs. Previous meta-heuristic solutions for VM consolidation problem consider only the current workloads on PMs when performing the placements, i.e., they treat the workloads as momentary events; this may lead to many future overloads and VM migrations (after placement) due to server oversubscriptions. To the best of our knowledge, none of the previous works that solve the VM consolidation problem using meta-heuristic algorithms deal the workload data as non-stationary stochastic processes. Hence, our work, solves the problem from a new point of view when compared with its counterparts.

III. Proposed System

In this paper, an energy efficient and SLA aware system is proposed for dynamic management of heterogeneous VMs and PMs in Cloud datacenters using GA optimization and stochastic processes. The system dynamically models the workload changes and adapts itself to the current workloads. The system model is shown in Fig. 4. The system is consisted of heterogeneous PMs on which heterogeneous VMs work. Virtual Machine Monitor (VMM) on each host is responsible for continuous monitoring of the host resource utilization, step detection, estimating utilization model, and cooperating with the Data Center Manager (DCM) by sending adjustment requests, so that the DCM can make appropriate decisions and issues controlling commands. To do this, DCM employs a SLA-aware VM placement algorithm using GA to find migration map and then sends out migration commands to VMMs.

A. *Non-stationary Workload Data Modeling*

In this study, we use the assumptions explained in [15] that is, real workloads are assumed not to be completely random, and thus they are independent stochastic processes which may be non-stationary. If we consider the CPU utilization of a VM_k as a random variable (represented as X_k), there are n independent random variables on a PM. Loads on different VMs are not identically distributed, but it can be shown that the Lindeberg-Feller Central Limit Theorem condition holds. Let σ_k^2 be the variance of X_k and $s_n^2 = \sum_{k=1}^{n} \sigma_k^2$, then the sequence of independent random variables on a PM satisfies the following condition [37].

$$\max_{k=1,\dots,n} \frac{\sigma_k^2}{s_n^2} \to 0, \ as\ n \to \infty \qquad (1)$$

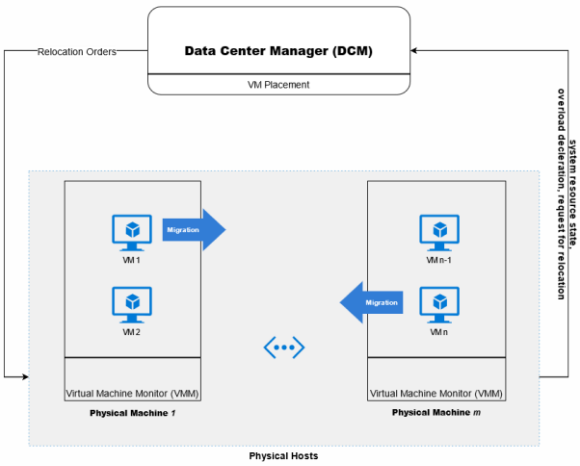

Fig. 4. High level view of the system model.

[DOI: 10.52547/itrc.14.2.14]

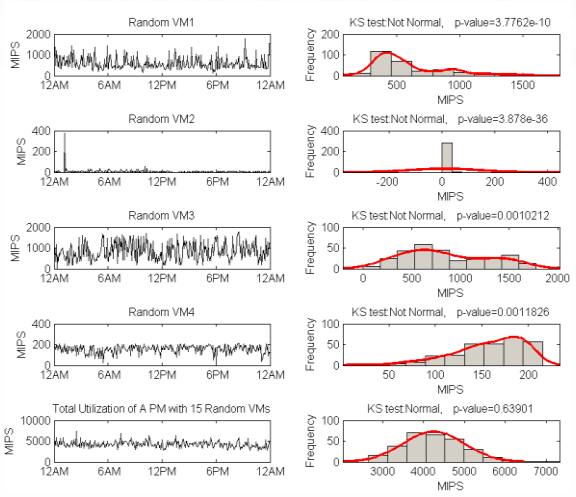

Fig. 5. Energy Consumption diagrams and histograms of four of the 15 virtual machines with non-normal consumption distributions randomly selected from the PlanetLab [38] data set and placed on a host, along with diagram and histogram of the physical host

Consumption diagrams and histograms of four of the 15 virtual machines with non-normal consumption distributions randomly selected from the PlanetLab [38] data set and placed on a host, along with diagram and histogram of the physical host

Thus, the workload on a PM has a normal distribution. We illustrated an example of this in Fig. 5: 15 real loads with non-normal distribution (based on the Kolmogorov-Smirnov (KS) test) from the PlanetLab [38] data set were randomly placed on a host, and the normality of the load data was tested using the KS test; the total utilization on the host has a normal distribution.

However, real-world workload data may be non-stationary, and one of the methods used to deal with non-stationary data is sliding window method [16]. The problem with a fixed-length sliding window is that it has to be tested for different lengths in the problem to find the right length for it. To solve this problem, we have provided a solution to dynamically determine the size of the sliding window. Since the data distribution is considered to be normal, we use two sample Student's t-test with unequal sample sized and unequal variances to detect the significant changes in the workloads on PMs. As discussed by Carter and Cross [39], we can use two-sample Student's t-test to find whether the means of two set of observations are significantly different. The *t* statistic to test whether the sample means are different is calculated as follows,

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \qquad (2)$$

Where n_1 is the number of points in the initial samples (in our experiences, 30 points after the current detected change point), n_2 is the count of samples from the current step point (excluding the initial samples) to the current time. s_1, s_2 are sample variances and $\bar{x}_1$, $\bar{x}_2$ are sample means. The degree of freedom is calculated as follows.

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}} \qquad (3)$$

Fig. 6 shows the proposed algorithm used for finding the dynamic size of the sliding window on each host. By finding the length of the sliding window (l), the distribution parameters are estimated using the maximum likelihood estimation (MLE) method. Therefore, the overload probability of physical hosts can be calculated by having load distribution parameters. The obtained values of the probabilities are used in the VM placement solution, which is explained in the next section.

```
   Input: Workload on a host: load
   Output: length of sliding window: l

1  Function findSlidingWindowLength(load) begin
2    point ← 0, i ← 0
3    while (i < load.length() ) do
4      x1 ← SubArray(load, point, point + n1))
5      i = point + n1
6      for (j=i+1 to load.length()) do
7        x2 ← SubArray(load, i + 1, n1 + j))
8        stepDetected ← t_test.isSignificantChange(x1, x2)
9        if (stepDetected) then
10         point ← i + 1 + n1 + j
11         break
12       end if
13     end for
14   end while
15   l = load.length()-point
16   return l
   end Function
```

Fig. 6. The proposed algorithm for sliding window length detection.

```
   Input: PM list: PmList
   Output: MigrationMap: Map

1  Function Consolidation(PmList) begin
2    overUtilizedPMList ← new List < PM >
3    vmsToMigrate ← new List < VM >
4    foreach (pm in PmList) //finding overloaded PMs
5      slidingWindowLen←findSlidingWindowLenght (pm.load)
6      if(overloadProbaility(pm.load[slidingWindowLen]> th_o)
7        overUtilizedPMList.Add(pm)
8      else if (LocalRegression(pm.load) > pm.Capacty)
9        overUtilizedPMList.Add(pm)
10     end if
11   end foreach
12   foreach (pm in overUtilizedPMList) //VM selection from overloaded PMs
13     foreach (vm in pm.VmList order by descending MMT)
14       vmsToMigrate ← vm
15       pm.VmList.remove(vm)
16       if (pm is not overloaded) // after removing vm from pm.VmList
17         break;
18       end if
19     end foreach
20   end foreach
21   destPMList ← PmList.Remove (overUtilizedPMList)
     // for all PMs (when some VMs are assigned to) the sliding window length is
     // calculated and GA is executed based on Eq. 3 fitness
22   migrationMap ← SlaAwareGaVmPlacement(vmsToMigrate, destPMList)
     // All the PMs that are not overloaded and their residing VMs
     // (according to migrationMap) are given to the Placement algorithm
23   underloadMigMap ←getMigMapFromUnderUtilizedPMs(migrationMap,
                       destPMList)
24   migrationMap.addAll(underloadMigMap);
25   return migrationMap
26 end Function
```

Fig. 7. The proposed consolidation algorithm

[DOI: 10.52547/itrc.14.2.14]

B. Stochastic process-based GA for VM Consolidation

***SLA-aware GA for VM Placement*:** The problem of VM placement on servers have always been a challenge for Cloud data centers. The basic idea of different policies is based on mapping the VM placement problem to bin packing problem with the goal of reducing the number of active servers to reduce energy consumption.

Here, a new GA-based VM placement algorithm is proposed which is SLA aware. As stated before, the workloads are considered as non-stationary stochastic processes, so the GA placement is enabled to care about SLAs. The optimization problem is considered as the placement of n VMs on minimum number of hosts while keeping the overload probability below a safety threshold. The optimization problem can be summarized as follows,

$$Minimize\ number\ of\ hosts \tag{4}$$

$$where:$$

considering current stationary utilization obtained from Fig. 6 algorithm

$$p_o \leq th_s,$$

$$\sum_{h=1}^{m} h_j = 1. \quad \forall j \in \{1, \dots, n\}$$

Where h_j=1 if the VM j is allocated to the host h, p_o is the overload probability of the host considering its last l utilization observations which is obtained using the algorithm shown in Fig. 6, and th_s is the safety threshold determines the importance of SLAs. Choosing smaller values for th_s, reduces future SLA violation probability on a host. Note that, lower SLA violation increases the potential of higher energy consumption in the data center. However, considering SLA violation probability before the placements has benefits that cannot be omitted: It reduces total SLA violation in the system by having fewer overloading hosts and also fewer migration due to overload (which also imposes SLA violation). In other words, as shown in Experiments section, the simultaneous optimization of energy consumption and SLA violation which is evaluated by ESV metric (discussed in section IV) significantly improves by employing the proposed idea in the GA-based placement algorithm. To encode the problem, each VM is considered as a gene, and thus a chromosome consists of n (number of VMs) gens. The value of a gene is a number between 1 and m (number of PMs), which determines the PM that will host the corresponding VM. Simple mutation and crossover (uniform) functions are used similar to the ones described in [35, 36].

Our GA-based VM placement has various advantages: by considering the utilization of after placement, and then by employing Fig. 6 algorithm, the proposed placement algorithm uses the best length of historical data for estimating utilization distribution parameters. More importantly, the proposed algorithm does not let VM placements which lead to overload probability over the predefined safety threshold th_s, which is also calculated based on the current utilization process. As a result, the proposed method avoids decision making based on momentary conditions by considering workloads as stochastic processes, thus it reduces both energy consumption and SLA violations simultaneously. In summary, the use of our placement algorithm leads to:

i) Less energy consumption by selecting minimum set of active hosts
ii) Accurate overload probability estimations due to using adaptive approach for workload data modeling, and thus good SLA establishment.

Overload and Underload Host Detection: In this paper, according to the utilization process of a host, whenever the probability of overload becomes greater than a predefined threshold (th_o), we consider the host as overloaded. A local regression method is also used for short term utilization prediction.

For underload detection, all the hosts that are not considered as overloaded are given to the placement algorithm to see if the PMs can be considered as underloaded. Since our VM placement algorithm is SLA-aware, the advantages of our approach is that it does not decide based on instantaneous resource consumption, which prevents short-term shutdowns. Fig. 7 shows the algorithm of the proposed consolidation process.

IV. EXPERIMENTS

A. Experiments Setup

For the sake of the repeatability of the experiments CloudSim Toolkit [40] is used. The simulated data center includes 800 heterogeneous servers with two types of physical machines: 400 HP ProLiant ML110 G5 (2 cores x 2660 Mhz), 400 HP ProLiant ML110 G4 (2 cores x 1860 Mhz). Real workload data of the CoMon project [38] (i.e. PlanetLab workload data) is considered in the experiments. We randomly selected 450 VMs from the dataset. GA parameters are set like the ones presented in [36], $th_o = 0.05, th_s = 0.05$, and the significance level of the t-test is set to 0.

B. Performance Metrics

The main metrics considered for evaluating the efficiency of the proposed algorithms are defined briefly as follows.

OTF: If the demand of the CPU on a host exceeds its capacity, the SLA is violated. Overload Time Fraction (OTF) is used to calculate the SLA violation due to resource shortage on PMs, which is

$$OTF = \frac{1}{|PM|}\sum_{i=1}^{|PM|}\frac{T_{o_i}}{T_{a_i}} \tag{5}$$

Where |PM| is the PM counts and T_{o_i} is the total overload time. T_{a_i} is the total time that PM_i has been active.

PDM: Performance degradation by VMs due to Migration (PDM) which is calculated as follows.

$$PDM = \frac{1}{|VM|}\sum_{j=1}^{|VM|}\frac{C_{d_j}}{C_{r_j}} \tag{6}$$

[DOI: 10.52547/itrc.14.2.14]

Where |VM| is the total VM count; C_{d_j} is the performance degradation of VM_j caused by migration. C_{r_j} is the total CPU capacity in MIPS requested by VM_j during its lifetime the average reduction in performance (C_{d_j}) is assumed to be equal to 10% of CPU utilization during all migrations of VM_j [8, 11, 15].

SLAV: SLA Violation (SLAV) is a multi-parameter metric which considers both the OTF and PDM metrics, which previously defined in section III.

$$SLAV = OTF \times PDM \quad (7)$$

ESV: this metric is used for simultaneous evaluation of the optimization of energy and QoS.

$$ESV = Energy \times SLAV \quad (8)$$

C. Comparing with benchmark approaches

Here, we discuss the performance evaluation of our proposed system in comparison with benchmark approaches. The benchmark algorithms include the Heuristic-based Dynamic Server Consolidation (HDSC) approach proposed in [8], Energy and SLA efficient VM Consolidation (ESVMC) approach proposed in [27], Simple GA based VM placement (SGAVMP) algorithm proposed in [36], and Stochastic-based Dynamic Server Consolidation (SBDSC) approach proposed in [15]. HDSC, and ESVMC, SGAVMP are considered with a Local Regression (LR) method for overload host detection and SBDSC uses a stochastic process-based solution for overload host detection which is similar to the one described in this paper, but it works with static sliding window. All of the benchmark solutions use MMT policy as their VM selection algorithm and differ in VM placement and underload detection algorithms. HDSC is implemented in the CloudSim simulator by default. We implemented the algorithms of the other approaches in CloudSim and the input data are all considered the same, as mentioned in Section IV part A.

The results show that the usage of the dynamic approach for determining the length of the historical data for estimations has reduced the OTF values (Fig. 8), and its combination with our VM placement algorithm has led to a decreased number of migrations due to overload (Fig. 9) and a lower SLA violation in terms of the SLAV metric (Fig. 10). The main reason is that out proposed system has more accuracy in predicting overloaded hosts, and thus both OTF and SLAV reduces; another reason to the lower SLA violation is that our approach indirectly prevents unnecessary underload detections. The main reason that the number of migrations reduces is that our placement algorithm does not decide based on momentary information, in contrary to HDSC, ESVMC and SGAVMP.

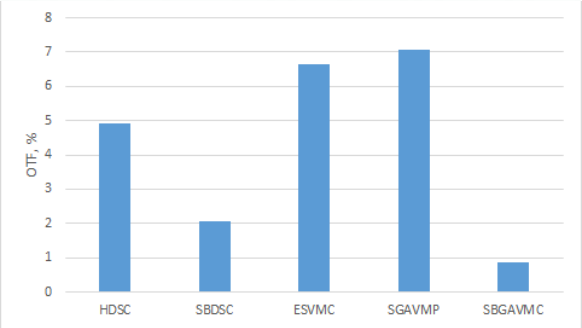

Fig. 8. OTF (%) values for different server consolidation policies.

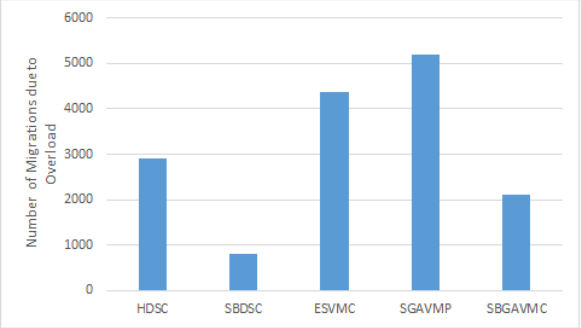

Fig. 9. Number of migrations due to overload for different server consolidation policies.

On the other hand, it can be seen in Fig. 11 that the proposed VM placement which is power-efficient and SLA-aware has led to low energy consumption in compression with its heuristic counterparts, which is about 9% lower than the ESVMC approach, but this method consumes more energy than the simple GA-based VM placement method because the SGAVMP is not SLA aware and it violates the SLAs more than other approaches (7.1% more overload time fraction than our approach). These indicates that the efficiency of our method is acceptable in terms of energy consumption. Finally, the proposed consolidation approach in this paper outperforms all the benchmark algorithms in reducing the total cost in the system, which is represented by the ESV metric as shown in Fig. 12. Results show that ESV value has improved about 45% when compared with the best of the benchmark approaches, i.e., SBDSC.

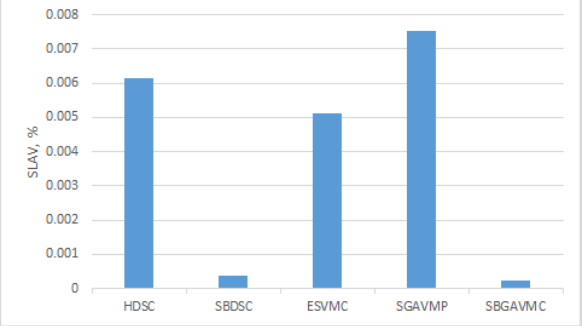

Fig. 10. SLAV (%) values for different server consolidation policies.

[DOI: 10.52547/itrc.14.2.14]

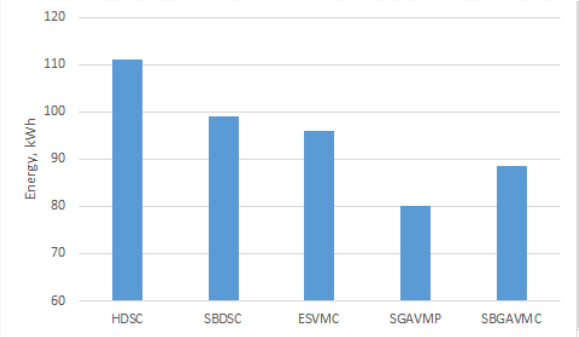

Fig. 11. Energy consumption of different server consolidation policies.

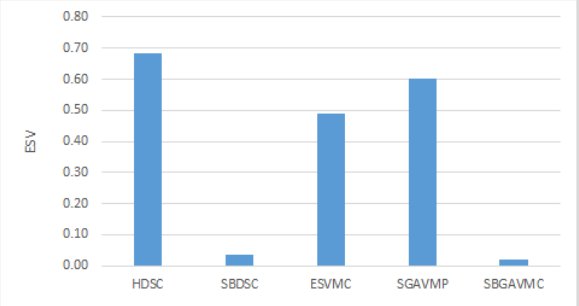

Fig. 12. ESV values for different server consolidation policies

V. Conclution and Future workks

In this paper, we presented a non-stationary workload adaptive server consolidation approach which is energy efficient and also has a high QoS. In this regard, we introduced an approach for modeling the utilizations of VMs and PMs based on the dynamic variations happens in utilization processes during time. In addition, a new VM placement algorithm was proposed which succeed in reducing the total energy consumption in the system while avoiding future SLAs violations. Our proposed VM placement approach outperforms its meta-heuristic counterparts, since unlike the previous works that usually make decisions based on the current workload or the predicted behavior of workloads, in this paper, we consider the workloads as stochastic processes, which enables the system to be SLA-aware (i.e., fewer overload condition happens) and avoid unnecessary underload detections (i.e., lower SLA violation due to migration).

The performance of our proposed algorithms were evaluated by the means of CloudSim simulation Toolkit and real workload data. The performance of the consolidation process has greatly improved when compared with benchmark approaches. The performance of the system has improved in SLA violations, ESV, and number of migrations due to overload. For future work, we want to further improve the proposed system by finding solutions for more energy efficient VM placement while keeping SLA violations as low as possible.

References

[1] Jennings, B. and R. Stadler, *Resource management in clouds: Survey and research challenges.* Journal of Network and Systems Management, 2015. **23**(3): p. 567-619.

[2] Beloglazov, A., J. Abawajy, and R. Buyya, *Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing.* Future generation computer systems, 2012. **28**(5): p. 755-768.

[3] Andrae, A.S. and T. Edler, *On global electricity usage of communication technology: trends to 2030.* Challenges, 2015. **6**(1): p. 117-157.

[4] Monshizadeh Naeen, H., E. Zeinali, and A. Toroghi Haghighat, *Adaptive Markov - based approach for dynamic virtual machine consolidation in cloud data centers with quality - of - service constraints.* Software: Practice and Experience, 2020. **50**(2): p. 161-183.

[5] Van Steen, M. and A.S. Tanenbaum, *Distributed systems*. 2017: Maarten van Steen Leiden, The Netherlands.

[6] Mishra, M., et al., *Dynamic resource management using virtual machine migrations.* IEEE Communications Magazine, 2012. **50**(9): p. 34-40.

[7] Clark, C., et al. *Live migration of virtual machines*. in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. 2005. USENIX Association.

[8] Beloglazov, A. and R. Buyya, *Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers.* Concurrency and Computation: Practice and Experience, 2012. **24**(13): p. 1397-1420.

[9] Voorsluys, W., et al., *Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation.* CloudCom, 2009. **9**: p. 254-265.

[10] Zolfaghari, R., et al., *An energy - aware virtual machines consolidation method for cloud computing: Simulation and verification.* Software: Practice and Experience, 2022. **52**(1): p. 194-235.

[11] Arianyan, E., H. Taheri, and V. Khoshdel, *Novel fuzzy multi objective DVFS-aware consolidation heuristics for energy and SLA efficient resource management in cloud data centers.* Journal of Network and Computer Applications, 2017. **78**: p. 43-61.

[12] Li, L., et al., *SLA-aware and energy-efficient VM consolidation in cloud data centers using robust linear regression prediction model.* IEEE Access, 2019. **7**: p. 9490-9500.

[13] Mustafa, S., et al., *Sla-aware best fit decreasing techniques for workload consolidation in clouds.* IEEE Access, 2019. **7**: p. 135256-135267.

[14] Barthwal, V. and M.M.S. Rauthan, *AntPu: a meta-heuristic approach for energy-efficient and SLA aware management of virtual machines in cloud computing.* Memetic Computing, 2021. **13**(1): p. 91-110.

[15] Monshizadeh Naeen, H., E. Zeinali, and A. Toroghi Haghighat, *A stochastic process-based server consolidation approach for dynamic workloads in cloud data centers.* The Journal of Supercomputing, 2020. **76**(3): p. 1903-1930.

[16] Beloglazov, A. and R. Buyya, *Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints.* IEEE Transactions on Parallel and Distributed Systems, 2013. **24**(7): p. 1366-1379.

[17] Barroso, L.A. and U. Hölzle, *The case for energy-proportional computing.* Computer, 2007. **40**(12).

[18] Gao, Y., et al., *Service level agreement based energy-efficient resource management in cloud data centers.* Computers & Electrical Engineering, 2014. **40**(5): p. 1621-1633.

[19] Lee, Y.C. and A.Y. Zomaya, *Energy efficient utilization of resources in cloud computing systems.* The Journal of Supercomputing, 2012. **60**(2): p. 268-280.

[20] Usmani, Z. and S. Singh, *A survey of virtual machine placement techniques in a cloud data center.* Procedia Computer Science, 2016. **78**: p. 491-498.

[21] Bodik, P., et al., *A case for adaptive datacenters to conserve energy and improve reliability.* University of California at Berkeley, Tech. Rep. UCB/EECS-2008-127, 2008.

[DOI: 10.52547/itrc.14.2.14]

[22] Guenter, B., N. Jain, and C. Williams. *Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning*. in *2011 Proceedings IEEE INFOCOM*. 2011. IEEE.

[23] Zolfaghari, R. and A.M. Rahmani, *Virtual machine consolidation in cloud computing systems: Challenges and future trends.* Wireless Personal Communications, 2020. **115**(3): p. 2289-2326.

[24] Nathuji, R. and K. Schwan. *Virtualpower: coordinated power management in virtualized enterprise systems*. in *ACM SIGOPS Operating Systems Review*. 2007. ACM.

[25] Verma, A., P. Ahuja, and A. Neogi. *pMapper: power and* migration *cost aware application placement in virtualized systems*. in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. 2008. Springer-Verlag New York, Inc.

[26] Srikantaiah, S., A. Kansal, and F. Zhao. *Energy aware consolidation for cloud computing*. in *Proceedings of the 2008 conference on Power aware computing and systems*. 2008. San Diego, California.

[27] Arianyan, E., H. Taheri, and S. Sharifian, *Novel heuristics for consolidation of virtual machines in cloud data centers using multi-criteria resource management solutions.* The Journal of Supercomputing, 2016. **72**(2): p. 688-717.

[28] Kusic, D., et al., *Power and performance management of virtualized computing environments via lookahead control.* Cluster computing, 2009. **12**(1): p. 1-15.

[29] Bishop, G. and G. Welch, *An introduction to the Kalman filter*. Proc of SIGGRAPH, Course, 2001. **8**(27599-3175): p. 59.

[30] Horri, A., M.S. Mozafari, and G. Dastghaibyfard, *Novel resource allocation algorithms to performance and energy efficiency in cloud computing.* The Journal of Supercomputing, 2014. **69**(3): p. 1445-1461.

[31] Zhang, S., et al., *Burstiness-aware resource reservation for server consolidation in computing clouds.* IEEE Transactions on Parallel and Distributed Systems, 2016. **27**(4): p. 964-977.

[32] Jalalian, Z. and M. Sharifi, *A Survey on Task Scheduling Algorithms in Cloud Computing for Fast Big Data Processing.* International Journal of Information and Communication Technology Research, 2021. **13**(4): p. 28-35.

[33] Farahnakian, F., et al., *Using ant colony system to consolidate VMs for green cloud computing.* IEEE Transactions on Services Computing, 2015. **8**(2): p. 187-198.

[34] Li, H., et al., *Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing.* Computing, 2016. **98**(3): p. 303-317.

[35] Tang, M. and S. Pan, *A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers.* Neural Processing Letters, 2015. **41**(2): p. 211-221.

[36] Wu, G., et al. *Energy-efficient virtual machine placement in data centers by genetic algorithm*. in *International conference on neural information processing*. 2012. Springer.

[37] Papoulis, A. and S.U. Pillai, *Probability, random* variables, *and stochastic processes*. Fourth ed. 2002: Tata McGraw-Hill Education.

[38] Park, K. and V.S. Pai, *CoMon: a mostly-scalable monitoring system for PlanetLab.* ACM SIGOPS Operating Systems Review, 2006. **40**(1): p. 65-74.

[39] Carter, N.J. and R. Cross, *Mechanics of the kinesin step.* Nature, 2005. **435**(7040): p. 308-312.

[40] Calheiros, R.N., et al., *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms.* Software: Practice and experience, 2011. **41**(1): p. 23-50.

Hossein Monshizadeh received his Ph.D. in Computer Engineering in 2019. He also received his B.Sc. and M.Sc. degrees in Computer Engineering in 2010 and 2013, respectively. He is an assistant professor at the Islamic Azad University (IAU) of Neyshabur, Neyshabur, Iran. Currently, He is the head of the Department of Computer and Electrical Engineering at IAU-Neyshabur. His current research interests include Fog & Cloud Computing, Internet of Things, Computer Networks, Data Mining, and Stochastic Processes.

[DOI: 10.52547/itrc.14.2.14]