

Energy-Aware Adaptive Four Thresholds Technique for Optimal Virtual Machine Placement

A. R. Mohazabiyeh¹, KH. Amirizadeh²

¹Department of Computer Engineering, Bushehr Branch, Islamic Azad University, Bushehr, Iran

²Department of Computer Engineering, Garmsar Branch, Islamic Azad University, Garmsar, Iran

Article Info

Article history:

Received Feb 15, 2018

Revised May 17, 2018

Accepted May 24, 2018

Keyword:

Adaptive four-threshold

Cloud computing

Dynamic virtualization

Energy efficiency

SLA Agreement

ABSTRACT

With the increasing expansion of cloud data centers and the demand for cloud services, one of the major problems facing these data centers is the "increasing growth in energy consumption". In this paper, we propose a method to balance the burden of virtual machine resources in order to reduce energy consumption. The proposed technique is based on a four-adaptive threshold model to reduce energy consumption in physical servers and minimize SLA violation in cloud data centers. Based on the proposed technique, hosts will be grouped into five clusters: hosts with low load, hosts with a light load, hosts with a middle load, hosts with high load and finally, hosts with a heavy load. Virtual machines are transferred from the host with high load and heavy load to the hosts with light load. Also, the VMs on low hosts will be migrated to the hosts with middle load, while the host with a light load and hosts with middle load remain unchanged. The values of the thresholds are obtained on the basis of the mathematical modeling approach and the *K*-Means Clustering Algorithm is used for clustering of hosts. Experimental results show that applying the proposed technique will improve the load balancing and reduce the number of VM migration and reduce energy consumption.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

A. R. Mohazabiyeh,
Department of Computer Engineering,
Islamic Azad University,
Bushehr, Iran.
Email: alirezamohazabie@gmail.com

1. INTRODUCTION

Cloud computing is a model based on large computer networks, such as the Internet, which provides a new model for the supply, use and delivery of IT services and other computational resources for the use of the Internet. Cloud computing is now more focused, because people can access resources in a simple way. Cloud computing, unlike previous examples, offers infrastructure as a service (IaaS), a database as a service (PaaS), and software as a service (SaaS) [1], [2].

Cloud computing has emerged as a distinct way to lease IT infrastructure for everyday use in the short term. Despite cloud computing, companies can have high capacity at the same time without having to invest in new infrastructure, training novice personnel, or licensing new software. Cloud computing offers a special advantage for small and medium-sized businesses who want to completely outsource their data center infrastructure, or companies that want to load their capacity without imposing high cost of consuming large data centers.

The problem of high energy consumption in cloud data centers has become a serious problem due to large-scale construction, and on the other hand, such data centers that consume large amounts of electrical energy will result in high energy consumption and carbon dioxide emissions. The global data center's consumption of 2013 was reported to be over 4.35 GW in 2013, with an annual growth rate of 15%. The

issue of high power consumption of data centers has caused problems such as energy dissipation, low Return on Investment (ROI), system instability and more carbon dioxide emissions [3], [4].

Of course, most hosts in data centers are in low CPU usage. Barroso and Holzle performed a survey over half a year and found that most hosts in data centers operate at lower than 50% CPU utilization [5]. Bohrer and colleagues examined the problem of high energy consumption and achieved the same result. Therefore, data center power consumption is very necessary at the same time as minimizing SLA aggression [6].

In this paper, we propose an optimal placement algorithm for virtual machines based on energy-aware adaptive Four-Thresholds technique for reducing energy consumption and minimizing service level violations in cloud datacenters, and We verify the effectiveness of the proposed algorithms using the CloudSim toolkit.

The main parts of the paper are summarized as follows:

- a. In the proposed method of optimizing the virtual machine placement algorithm based on the Energy-Aware adaptive four thresholds, hosts in data center are classified into five categories according to their load. Virtual machines are transferred from the host with high load and heavy load to the hosts with light load and transferred from hosts with low load to the hosts with middle load, while Virtual Machines in the host with light load and moderate load remains unchanged.
- b. Presenting an Adaptive Four-Threshold Algorithm to Determine the Four Thresholds.
- c. The use of a virtual device selection approach and an allocation algorithm.
- d. Evaluating proposed algorithms with extensive simulation using the CloudSim tool.

The rest of this paper is organized as follows. In Section 2, the related work is discussed. Section 3 presents the power model, the transfer cost of VM migration, SLA violation metrics, and energy efficiency metrics. Section 4 proposes the proposed method, the four-step approach algorithm, the VM selection approach, and the VM deployment algorithm. Experiments and performance evaluation are presented in Section 5. Section 6 provides the conclusions of the paper.

2. RELATED WORKS

There is currently a lot of research that focuses on managing energy efficiency resources in cloud data centers. Zhou Zhou and *et al.*, presented an adaptive three-threshold energy awareness algorithm in 2016. The purpose of providing this algorithm is to properly accommodate a virtual device on a data center by reducing the level of service-level violation [7]. Nida Jin *et al.*, presented the Firefly algorithm FFO-EEVM in 2016. This algorithm is provided for energy optimization in data centers and virtual machine migration with energy consciousness [8]. Yang Qiang and *et al.*, (2013) introduced a multi-objective anion clone system algorithm for cloud computing virtual machines called VMPACS. The goal of this algorithm was to improve the power efficiency and resource utilization in a cloud computing environment by fitting the virtual machines in the data centers [9]. Karaboga and *et al.*, (2011) and George and *et al.*, (2013) did a research on colony bee algorithms called ABC. Bee based algorithms are modeled on the behavior of bees in the hive or outside it, especially their behavior in finding the source of food. The ABC algorithm among other algorithms has the best performance in terms of finding the right response and speed, and is also suitable for solving complex problems [10]. Mansur Murshad *et al.*, (2014) offered the AVVMC algorithm to balance resources across servers with various computing resources such as memory, processor and network outlet in order to minimize power consumption. This method produces a complex solution for complex problems like bin-packing and produces an optimal solution for regular paths [11]. Boya *et al.*, Provided a VM (single-threshold) (ST) deployment algorithm based on a combination of VM choices. The ST algorithm adjusts the same value for server CPU utilization to ensure that all Servers are below this value. It is known that the ST algorithm can save energy and reduce the aggression of the SLA, but the aggression remains high [12]. Beloglazov and Boya provide an efficient energy resource management system that includes the distributor, global manager, local manager and virtual machine monitor (VMM). Beloglazov *et al.*, consider a new DT (double threshold) VM algorithm to improve energy efficiency. DT tests the two thresholds so that the CPU utilizes all the hosts between the two thresholds, although energy consumption and SLA aggression for the DT algorithm should be reduced to a greater degree. Prior to that, Beloglazov and Boya proposed a double threshold adaptive VM placement algorithm to improve energy efficiency in data centers. However, energy consumption in datacenters remains high [13].

3. POWER MODEL, COST OF VM MIGRATION, SLA VIOLATION METRICS AND ENERGY EFFICIENCY METRICS

3.1. Power model

Energy consumption by servers in data centers is related to CPU, memory, disk, and bandwidth. Recent studies [7] have shown that, even if the DVFS method is used, the energy consumption by servers has a linear relationship with its CPU utilization. However, with the decrease of hardware price, multicore CPUs and memory with large-capacity are widely equipped in servers, and caused the conventional linear model not to be able to accurately determine the energy consumption of the servers. In order to deal with this problem, we use actual energy consumption data, which was suggested by SPEC power benchmark.

3.2. VM migration cost

Proper VM migration between servers can reduce energy consumption and SLA violation in data centers. excessive VM migration, of course, can negatively affect the performance of applications running on VMs. Voorsluys *et al.*, [14] investigated the problem of VM migration cost. Reduction in the performance of the VM can be expressed as follows:

$$C = K \int_{t_0}^{t_0+T_{mj}} u_j(t) dt \quad (1)$$

$$T_{mj} = \frac{M_j}{B_j} \quad (2)$$

Where C represents a decrease in the overall performance due to the VM j (virtual machine transfer cost), parameter k is the average coefficient of performance deviation caused by virtual machines (k value can be estimated about 0.1 (10%) of CPU utilization in terms of categories of web applications). The function $u_p(t)$ corresponds to the amount of processor utilization by VM j , the parameter t_0 is the start time of the transfer, T_{mj} is the completion time, M_j is the total memory used by VM j , and B_j represents the available bandwidth.

We have selected two servers equipped with dual-core processors. The main configuration of the two servers is as follows:

One of them is HP ProLiant G4 with 1.86GHz (dual-core), 4GB RAM and the other is HP ProLiant G5 with 2.66GHz (dual-core), 4 GB RAM. Energy consumption for the two servers at different load levels is presented in Table 1 [15].

Table 1. Power Consumption by the two Servers at different Load Levels in Watts

| Server | 0% | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|----------------|------|------|------|-----|------|-----|-----|-----|-----|-----|------|
| HP ProLiant G4 | 86 | 89.4 | 92.6 | 96 | 99.5 | 102 | 106 | 108 | 112 | 114 | 117 |
| HP ProLiant G5 | 93.7 | 97 | 101 | 105 | 110 | 116 | 121 | 125 | 126 | 133 | 135 |

3.3. SLA violation metrics

SLA violation is a very important factor for any VM migration algorithm. there are currently two methods for describing the SLA violation [16].

a. PDM (Overall loss of performance due to VM Migration). It is indicated in the equation:

$$PDM = 1/M \sum_{j=1}^M C_{dj}/C_{rj} \quad (3)$$

Where parameter M represents the number of virtual machines in the data center, C_{dj} is the estimate of the yield loss due to the transmission of VM j and C_{rj} correspond to the total capacity of the demanded processor by VM j during its lifetime.

b. SLATAH (SLA Violation Time per Active Host). It means the percentage of total SLA violation time, during which the CPU utilization by the active host has reached 100% and is shown by the following equation:

$$SLATAH = 1/N \sum_{i=1}^N T_{si}/T_{ai} \quad (4)$$

Where N denotes the number of hosts in the data center, T_{si} is the total time during which the processor utilization by the host i is 100% and raises the service level Agreement, T_{ai} is related to the time that the host

i is active. The logic of the service level agreement is that the operation of the active host CPU has experienced a 100% exploitation rate, and the virtual machines on the host cannot be provided by the processor demanding capacity.

Both PDM and SLATAH are two effective methods for independent assessment of the SLA violation. Therefore, the SLA violation is defined as the following equation [16]:

$$SLA = PDM \times SLATAH \quad (5)$$

3.4. Energy efficiency metric

Energy efficiency includes energy consumption and SLA violation. Improving energy efficiency means less energy consumption and less SLA violation in data centers. Therefore, the metric of energy efficiency is defined as :

$$E = 1/P \times SLA \quad (6)$$

Where E corresponds to the energy efficiency of a data center, P is the energy consumption of a data center, and SLA represents the SLA violation of a data center. Equation (6) shows that the higher the E , the greater the energy efficiency[7].

4. PROPOSED METHOD, ADAPTIVE FOUR-THRESHOLD ALGORITHM, VM SELECTION APPROACH, AND VM ALLOCATION ALGORITHM

4.1. Proposed method

VM migration is an effective method for improving energy efficiency in data centers. Of course, there are several key issues that need to be addressed:

(1) When it is assumed that a host has a heavy load, a number of virtual machines from the host must be transferred to another host; (2) when we know that a host must be moderately loaded or lightly loaded, we decide to keep all virtual machines in this host unchanged; (3) when we know that a host must be low-loaded, all virtual machines in the host must be transferred to another host; (4) selecting a VM or more VMs that should be migrated from the heavily loaded; (5) finding a new host to accommodate migrated VMs from heavily loaded or little-loaded hosts.

In the proposed method, we automatically select four thresholds, T_{low} , T_{light} , T_{middle} and T_{heavy} for solving problems ($0 \leq T_l < T_{li} < T_m < T_h \leq 1$), which causes Data center hosts be divided into five categories : hosts with low load, hosts with light load, hosts with middle load, hosts with high load and hosts with heavy load. The value of these four thresholds are utilized automatically using the threshold algorithm according to the load.

In the host, the processor utilization rate is less than T_l ($U < T_l$), in hosts with light load between T_l and T_{li} ($T_l < U < T_{li}$), in hosts with middle load between T_{li} and T_m ($T_{li} < U < T_m$), at high load between T_m and T_h ($T_m < U < T_h$), and in the heavy load the processor utilizes more than T_h ($U > T_h$). Virtual machines are transferred from the host with high load and heavy load to the hosts with light load, and from hosts with low load to the hosts with middle load and hosts with low load go to sleep, while the host computer is hosted with light load and host middle load remains unchanged. Figure 1 shows the flowchart of the proposed method.

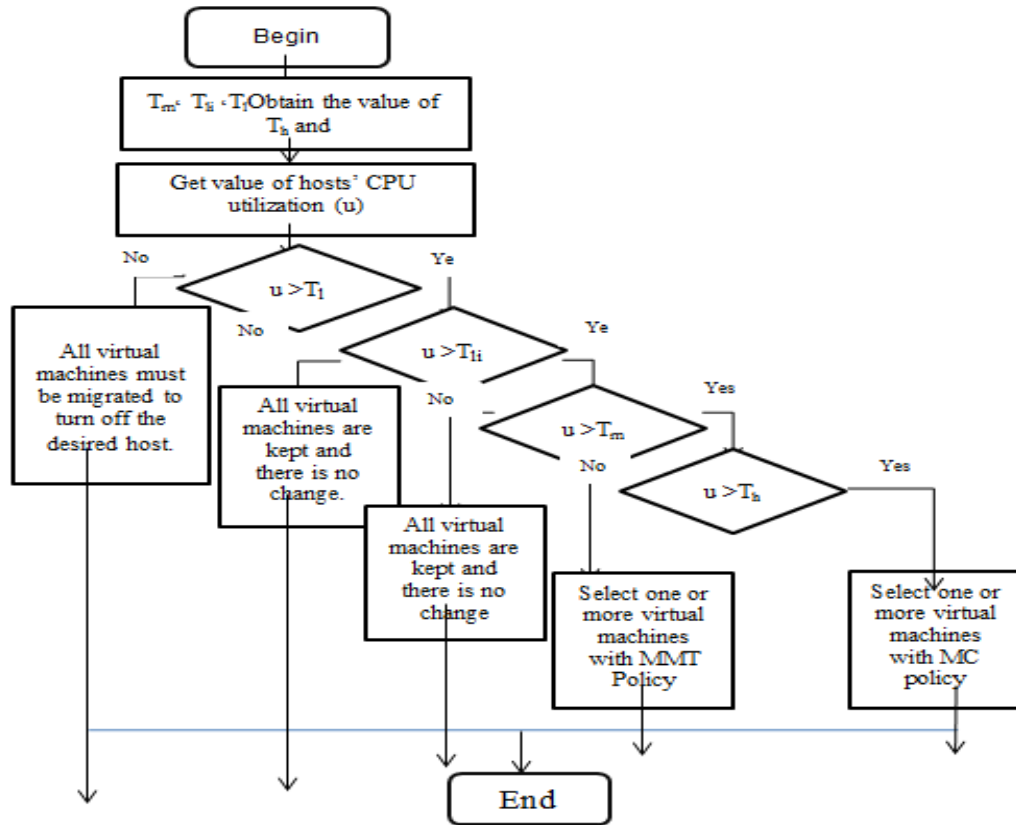


Figure 1. Flow chart of the Proposed Method

4.2. Adaptive threshold algorithm (detecting overhead hosts)

As discussed in Section 4.1, what are the threshold values of T_l , T_{li} , T_m and T_h ? To solve this Problem, K-Means Algorithm -Average-Median Absolute Deviation is proposed.

4.2.1. KAM (K-means clustering algorithm-average-median absolute deviation)

For the univariate data set of a variable $V_1, V_2, V_3 \dots V_n$ (V_i is CPU utilization of a host at time i , and the size of n can be determined by experimental value), the KAM algorithm uses the K-means clustering algorithm at first for dividing the data set ($V_1, V_2, V_3 \dots V_n$) into m groups (G_1, G_2, \dots, G_m) (the size of m can be determined by experimental value, and in this paper, $m = 5$), where $G_k = (V_{jk-1+1}, V_{jk-1+2}, \dots, V_{jk})$, for all $1 \leq k \leq 5$, and $0 = j_0 < j_1 < j_2 < \dots < j_5 = n$. Subsequently, KAM gets the average value of each group, formalized as follows [24]:

$$GAK = (V_{j+1+1}, V_{j+2+2} \dots V_{jk}) / (jk-jk-1) \tag{7}$$

For all $1 \leq k \leq 5$. Then, KAM gets the Median Absolute Deviation (MAD) of ($G_{A1}, G_{A2} \dots G_{A5}$). Therefore, the MAD is defined as follows:

$$MAD = \text{median}_{Ap} (|GAP - \text{median}_{Aq} (GA)|) \tag{8}$$

Where $A_1 \leq A_p \leq A_5$ and $\text{median}_{Aq} (GAq)$ are the average value of GAq . Finally, the four thresholds (T_l , T_{li} , T_m and T_h) in the proposed method can be defined as follows:

$$T_l = 0.5 (1 - r \times MAD) \tag{9}$$

$$T_{li} = 0.7(1 - r \times MAD) \tag{10}$$

$$T_m = 0.9 (1 - r \times MAD) \tag{11}$$

$$T_h = 1 - r \times \text{MAD} \quad (12)$$

That $r \in \mathbb{R}^+$ represents a parameter of the algorithm that determines how the system will consolidate the virtual machine. For example, the more r , the greater the energy consumption, but in the consolidation of the virtual machines will result in less violations of the service level agreement. In the proposed model, we have considered experiments to simulate the value of r to be 5. The mean absolute error complexity is the mean average clustering algorithm K , $O(m \times n \times t)$, where m , n , and t are the number of groups, data size and the number of repetitions, respectively.

The value of $(T_l, T_{li}, T_m$ and $T_h)$ also varies according to the continuous change of V_i ($i = 1, 2, 3 \dots n$). As a result, the mean absolute error of the mean clustering algorithm K is an adaptive four-threshold algorithm. When the workloads are dynamic and unpredictable, the absolute error of the middle mean of the mean clustering algorithm K creates a higher energy efficiency by setting the value $(T_l, T_{li}, T_m$ and $T_h)$ (in Comparison with a fixed-threshold algorithm) [7].

4.3. Virtual machine selection approaches

As described in the previous section, a number of virtual machines on high load and heavy load host should be migrated to another host with light load. Which virtual machine should be transferred? In general, the amount of host usage of the processor and the size of the memory will affect its energy efficiency; Therefore the approaches (MMT, MC) will be addressed in this section.

4.3.1. Minimum migration time (MMT)

The transition time of a virtual machine will vary according to its different memory sizes. A virtual memory device with less memory means less migration time under the same spare network bandwidth. For example, a VM with 16 GB memory may have a transfer time equal to 16 times the device with 1 GB of memory. It's crystal clear that selecting the VM with 16 GB of memory or the VM with 1 GB of memory greatly affect energy efficiency of data centers. Therefore, if the host has a high load, this approach will select the virtual machine with the least amount of memory in comparison with other virtual transfer devices to the host for transfer. The above method chooses virtual device u to have the following conditions [17], [19].

$$\text{RAM}(u) \leq \text{RAM}(v), \forall v \in \text{VM}_i \quad (13)$$

Where VM_i means the set of VMs assigned to host i and $\text{RAM}(u)$ is the amount of memory currently used by the VM u .

4.3.2. Maximum correlation (MC)

The maximum correlation procedure is based on the proposed idea by Verma. The idea is that, if there is a high correlation between applications running on a server, it is more likely that the overloading occurs on the server. Based on this idea, those virtual machines should migrate that have the highest correlation of CPU consumption with other virtual machines. So, if a host has heavy load, machines that use the most resources on the host (use processor resources and memory more than others and known as a large virtual machine) is chosen for displacement. The above approach chooses the virtual machine VM u if it satisfies the following condition [17], [18].

$$\text{CPU}(u) + \text{RAM}(u) \leq \text{CPU}(v) + \text{RAM}(v), \forall v \in \text{VM}_i \quad (14)$$

Where VM represents the set of virtual machines assigned to host i , and the $\text{CPU}(u)$ and $\text{RAM}(u)$ are the amount of memory and processor which are used by VM u currently.

4.4. VM deployment algorithm (source allocation algorithm)

In the proposed method for selecting the best host for VM embedding, we use Power Aware Best Fit Decreasing Algorithm (PABFDA). The algorithm's method is that at first, it checks the hosts list and verify whether a host has extensive load using the additional load detection algorithm. Then, if the host has an overload, the algorithm applies the virtual machine selection policy to select the virtual machines to be migrated from the host. When a list of virtual machines that are migrated from hosts with overload is built, the algorithm for placing the virtual machine will find a new location for virtual machines that could be migrated [19].

The second step of the algorithm is to find the low load hosts and to place virtual machines from these hosts to another host. The algorithm returns a map of a combination of migratory virtual machines that

contains information about the insertion of a new virtual machine from the virtual machines that are selected for migration from both overload and low load hosts. The complexity of the algorithm is N^2 , where N is the number of hosts.

In this algorithm, input is the host list and the list of virtual machines, and output is the allocation of virtual machines.

The process of algorithm operation is expressed in stages as follow:

- a. Arrange a list of virtual machines from virtual machines to reduce the processor efficiency.
- b. For each virtual machine in the list of virtual machines, allocate the minimum power to the hosts as the maximum power.
- c. For each host in the host list, if the host has sufficient resources for virtual machines, it evaluates the power of virtual machines and hosts. If the power is less than the minimum power, the dedicated host is the current host and the power of the virtual machines and the host is low.
- d. If the dedicated host is not empty, then the virtual machine will be assigned to another device [18], [20].

5. EXPERIMENTS AND PERFORMANCE EVALUATION

5.1. Experiment setup

In this research, the CloudSim simulator has been used to simulate algorithms, and the scenario presented in [45] is used to simulate algorithms. In this research, three commonly used methods namely MAD, IQR, 3th, are simulated for the "When is migration time" issue and the three widely used methods namely MMT, RS, and MC are simulated for the "Which virtual machine to be selected for migration" issue and is compared in several scenarios.

5.2. Simulated scenario profile

In this scenario, a data center is simulated by 800 heterogeneous physical nodes. The physical nodes used in this simulation are made up of servers from the HP ProLiant ML110 G4 and the HP ProLiant ML110G5 servers. The HP ProLiant ML110 G4 servers are powered by 1860 MIPS (Millions of Instructions Per Second) and HP ProLiant ML110G5 servers use 2660 MIPS. Both models have 1 GB/s bandwidth. Each node has 8 GB of memory and one terabyte of storage space. The energy consumption of a physical host is from 175 watts with zero percent of processor usage up to 250 watts with 100 percent CPU usage. Each virtual machine needs a processor core with 2500, 2000, 1000, or 500 million orders per second, 128 MB of RAM and 1 GB of storage space. The user records request for the supply of 1195 heterogeneous virtual machines that simulates the total data center capacity. Each of the tests is executed 10 times, and the current results were based on average values. Table 2 shows the characteristics of the servers used in the experiment.

Table 2. The Characteristics of the Servers used in the Experiment

| Server | CPU | RAM | BW | HDD | VM | PM |
|----------------|-----------|-----|--------|-----|------|-----|
| HP ProLiant G4 | 1860 MIPS | 8GB | 1GB/PS | 1TB | 1195 | 800 |
| HP ProLiant G5 | 2660MIPS | 8GB | 1GB/PS | 1TB | 1195 | 800 |

In this section, we will simulate the proposed algorithm. at first, we describe the details of the procedure and introduce the software used for simulation, then we will examine the data and how to simulate and introduce the data set. In the end, the proposed method is compared with related work and evaluations are carried out and the results are discussed. Simulator software provides an opportunity for researchers to test their ideas. It also speeds up tests and reduces costs. Here we use the simulation algorithm to model the cloud and implement the proposed algorithm. In the following chapter, we describe how to simulate. Because of the benefits of the CloudSim tool such as supporting the Dynamic Source Request and Modeling Virtual environments rules, we chose it as the simulation tool for our experiments.

5.3. Simulation and analysis results

5.3.1. Check energy consumption

Table 3 shows the energy consumption in different algorithms. To ensure that the results from the improvement of the proposed method are not accidental, we tested the algorithm in ten rounds of implementation. The average execution of ten rounds in the last two lines is shown in KW/h and W/H units. As it can be seen, the proposed algorithm has the lowest average execution rate of 393 watts per hour, which

indicates an improvement in energy consumption in comparison with the three-threshold algorithm which its average execution rate is 485 watts per hour.

Table 3. The Energy Consumption in the Proposed Method and other Algorithms implemented in ten times

| | Round 1 | Round 2 | Round 3 | Round 4 | Round 5 | Round 6 | Round 7 | Round 8 | Round 9 | Round 10 | Avg (Kw/H) | W/H |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|------------|-----|
| Proposed | 0.5 | 0.01 | 0.53 | 0.66 | 0.61 | 0.05 | 0.12 | 0.55 | 0.33 | 0.57 | 0.393 | 393 |
| IQR/MMT | 0.43 | 0.43 | 0.44 | 0.44 | 0.41 | 0.65 | 0.15 | 0.53 | 0.66 | 0.58 | 0.472 | 472 |
| IQR/MC | 0.56 | 0.34 | 0.55 | 0.18 | 0.54 | 0.3 | 0.64 | 0.59 | 0.29 | 0.32 | 0.431 | 431 |
| IQR/RS | 0.42 | 0.45 | 0.44 | 0.42 | 0.43 | 0.65 | 0.66 | 0.55 | 0.42 | 0.34 | 0.478 | 478 |
| MAD/MMT | 0.44 | 0.39 | 0.41 | 0.4 | 0.45 | 0.36 | 0.39 | 0.62 | 0.53 | 0.3 | 0.429 | 429 |
| MAD/MC | 0.57 | 0.34 | 0.34 | 0.45 | 0.28 | 0.13 | 0.6 | 0.58 | 0.61 | 0.16 | 0.406 | 406 |
| MAD/RS | 0.42 | 0.4 | 0.43 | 0.44 | 0.45 | 0.45 | 0.28 | 0.47 | 0.52 | 0.55 | 0.441 | 441 |
| 3th/RS | 0.57 | 0.54 | 0.36 | 0.6 | 0.2 | 0.69 | 0.07 | 0.66 | 0.46 | 0.55 | 0.47 | 470 |
| 3th/mc | 0.67 | 0.4 | 0.54 | 0.64 | 0.67 | 0.43 | 0.56 | 0.39 | 0.4 | 0.15 | 0.485 | 485 |

In Figure 1, energy consumption has been shown by various algorithms. As shown in the diagram, the proposed method has the lowest energy consumption of 393 watts per hour. The MAD method as an overhead diagnostic algorithm and MC algorithm are ranked second with 406 watts per hours. The proposed algorithm has been operating around 3.5% better than the MAD/MC method, which works better than other algorithms and saves energy.

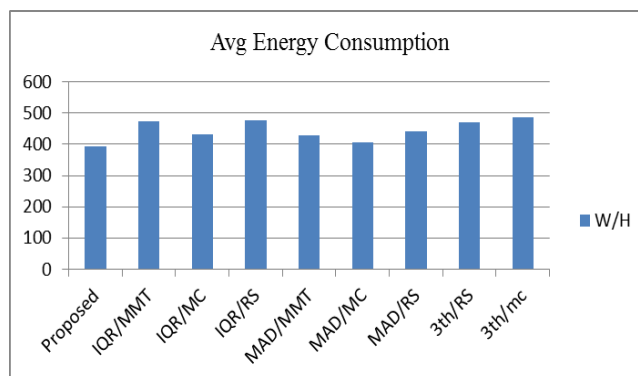


Figure 1. Average Energy consumption in different algorithms

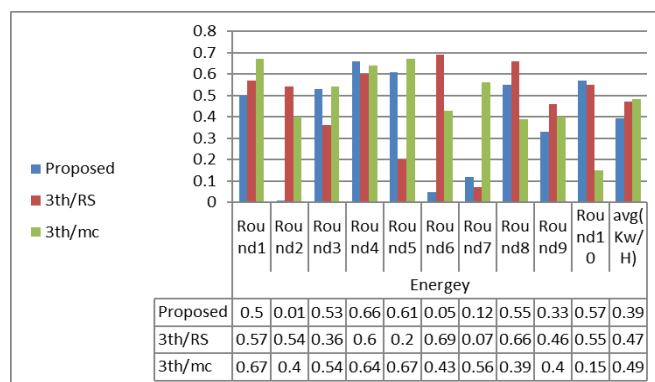


Figure 2. Comparison of performance of proposed algorithm and three thresholds for energy consumption in ten runs

As shown in Table 4, the proposed algorithm has an average of 96.8% energy consumption and around 3.5% performance improvement in comparison with the MAD/MC method; 81% energy consumption and 19% performance improvement in comparison with 3th/mc three-way algorithm; 83.6% energy consumption and 16.4% performance improvement in comparison with 3rd/RS.

Table 4. Comparison of the Energy Consumption Improvement of the Proposed Algorithm with other tested Algorithms

| Proposed | Energy Consumption Proposed /algorithms | Utilization Proposed /algorithms |
|----------|---|----------------------------------|
| IQR/MMT | % 83.3 | % 16.7 |
| IQR/MC | % 91.2 | % 8.8 |
| IQR/RS | % 82.2 | % 17.8 |
| MAD/MMT | % 91.6 | % 8.4 |
| MAD/MC | % 96.8 | % 3.2 |
| MAD/RS | % 89.1 | % 10.9 |
| 3th/RS | % 83.6 | % 16.4 |
| 3th/mc | % 81 | % 19 |

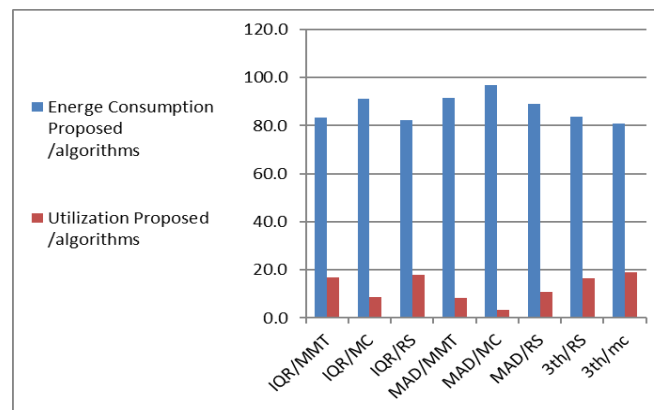


Figure 3. Comparison of the energy efficiency improvement of the proposed algorithm with other tested algorithms

5.3.2. Investigating SLA violation

Table 5 shows the degree of SLA violations (service-level-agreement) in different algorithms. The proposed algorithm with an average of 3.466 in the runs has the least violation in comparison with other algorithms, and the MAD / MMT algorithm with a mean of 5.455 has the highest violation. The proposed algorithm improved the performance with an average of 4.902 in comparison with the three-threshold algorithm.

Table 5. SLA Violation

| | Round 1 | Round 2 | Round 3 | Round 4 | Round 5 | Round 6 | Round 7 | Round 8 | Round 9 | Round 10 | AVG |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|-------|
| Proposed | 5.61 | 0 | 5.13 | 1.54 | 2.49 | 3.36 | 3.4 | 4.35 | 4.99 | 3.79 | 3.466 |
| IQR/MMT | 5.51 | 6.16 | 5.63 | 4.01 | 7.37 | 1.31 | 6.34 | 4.31 | 1.96 | 3.58 | 4.618 |
| IQR/MC | 3.88 | 4.88 | 3.66 | 4.23 | 3.36 | 5.24 | 2.34 | 2.79 | 3.37 | 5.77 | 3.952 |
| IQR/RS | 6.32 | 4.55 | 5.25 | 6.62 | 6.67 | 2.05 | 1.38 | 3.45 | 5.19 | 5.51 | 4.699 |
| MAD/MMT | 5.36 | 7.11 | 6.46 | 7.57 | 4.39 | 6.21 | 5.39 | 2.38 | 4.29 | 5.39 | 5.455 |
| MAD/MC | 2.8 | 5.72 | 5.61 | 5.06 | 5.65 | 6.33 | 3.14 | 2.8 | 6.84 | 5.52 | 4.947 |

| | Round 1 | Round 2 | Round 3 | Round 4 | Round 5 | Round 6 | Round 7 | Round 8 | Round 9 | Round 10 | AVG |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|-------|
| MAD/RS | 7.52 | 7.3 | 5.62 | 5.08 | 5.33 | 4.89 | 6.25 | 4.47 | 4.23 | 3.47 | 5.416 |
| 3th/RS | 4 | 4.88 | 7.27 | 3.56 | 7.53 | 1.29 | 8.14 | 1.76 | 5.87 | 4.8 | 4.91 |
| 3th/mc | 1.7 | 6.57 | 4.6 | 2.76 | 1.96 | 6.17 | 4.24 | 6.42 | 6.25 | 8.35 | 4.902 |

Figure 4 display the extent of the violation of the service level agreement. As outlined in this figure, the proposed method has the lowest level of SLA breach.

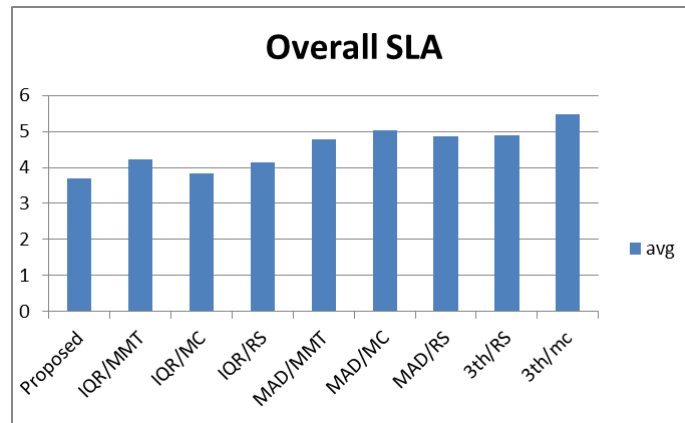


Figure 4. Average SLA Violation

Figure 5 shows the rate of changes of the proposed algorithm in SLA Violations in ten runs. It's crystal clear that it has the least change of service violations in comparison with other algorithms. As we can see the Figure 6, the three-threshold algorithm has the most changes of the service violation.

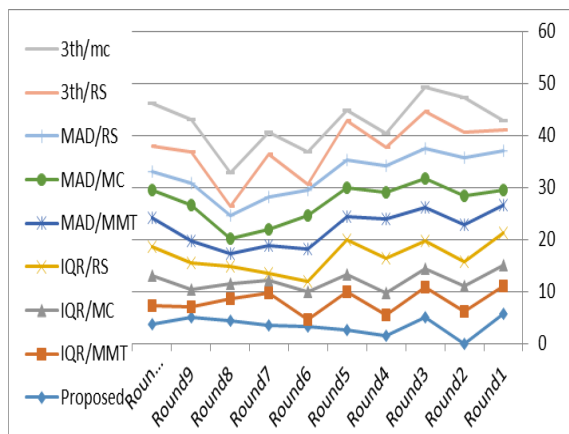


Figure 5. The amount of the proposed algorithm changes in SLA Violation in ten runs

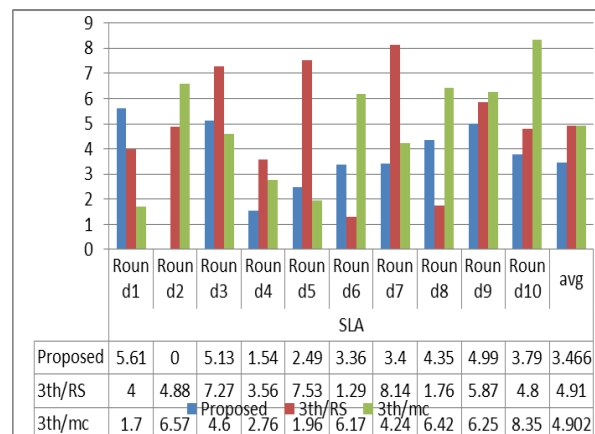


Figure 6. Comparison of the performance of the proposed algorithm and the three –thresholds algorithm for SLA Violation

As shown in Table 6, the proposed algorithm in comparison with the IQR / MC method had 87.7% of the SLA Violation , which operated about 0.49% improvement in performance and 12.3% of optimal performance, respectively, with respect to the three-thresholds algorithms 3th/mc, 70.7% violations of agreement and 29.3% improvement in performance and 3th/RS, 70.6% violations of agreement and 29.4% improvement in performance.

Table 6. The Percentage of Improvement of SLA Violation of the Proposed Algorithm in comparison with other Algorithms

| Proposed | Violation SLA Proposed /algorithms | Utilization SLA Proposed /algorithms | Performance improvement percentage/algorithms |
|----------|------------------------------------|--------------------------------------|---|
| IQR/MMT | %75.1 | %24.9 | 1.152 |
| IQR/MC | %87.7 | %12.3 | 0.486 |
| IQR/RS | %73.8 | %26.2 | 1.233 |
| MAD/MMT | %63.5 | %36.5 | 1.989 |
| MAD/MC | %70.1 | %29.9 | 1.481 |
| MAD/RS | %64.0 | %36.0 | 1.95 |
| 3th/RS | %70.6 | %29.4 | 1.444 |
| 3th/mc | %70.7 | %29.3 | 1.436 |

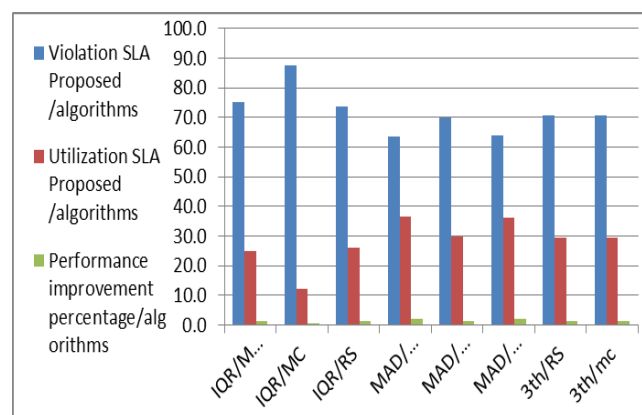


Figure 7. Comparing SLA Violation of the proposed algorithm with other tested algorithms

6. CONCLUSIONS AND FUTURE WORK

In Section 5, we simulated the proposed method. We tested our method with datasets. The results indicate an improvement in the performance of the virtual machine deployment algorithm. By examining the graphs in Section 5, the proposed method had the lowest energy consumption of 393 watts per hour, and had the least violation in terms of the level of SLA Violation. The proposed method succeeded in reducing energy consumption by about 3.5% and preventing violations of service-level agreement, which has had 29.3% and 12.3% performance improvement in comparison with three-thresholds algorithm and IQR/MC algorithm, respectively and has had around 1.4 and 0.49 performance improvement and has prevented violations of the SLA Violation.

In the future, our future work will offer the use of fuzzy algorithms for obtaining thresholds, changing the thresholds and matching or constanting the values of these thresholds, or using resource allocation algorithms and selecting a virtual machine to improve performance and reduce the energy of data centers and improvement of service level agreement.

REFERENCES

- [1] Khazaee, H., *et al.*, "Analysis of a Pool Management Scheme for Cloud Computing Centers", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 849-861, 2013.
- [2] J. Carretero, J., and Blas, J. G., "Introduction to Cloud Computing: Platforms and Solutions", *Cluster Computing*, vol. 17, no. 4, pp. 1225-1229, 2014.
- [3] Rincon, D., *et al.*, "A Novel Collaboration Paradigm for Reducing Energy Consumption and Carbon dioxide Emissions in Data Centres", *The Computer Journal*, vol. 56, no. 12, pp. 1518-1536, 2013.
- [4] Wang, L., and Khan, S. U., "Review of Performance Metrics for Green Data Centers: A Taxonomy Study", *The Journal of Supercomputing*, vol. 63, no. 3, pp. 639-656, 2013.
- [5] Barroso, L. A., and Hölzle, U., "The Case for Energy-proportional Computing", *Computer*, vol. 40, no. 12, 2007.

- [6] Ohrer, P., *et al.*, “The Case for Power Management in Web Servers”, In *Power Aware Computing*, pp. 261-289, Springer, Boston, MA, 2002.
- [7] Zhou, Z., *et al.*, “Virtual Machine Placement Algorithm for both Energy-awareness and sla Violation Reduction in Cloud Data Centers”, *Scientific Programming*, 2016, 15.
- [8] Kansal, N. J., and Chana, I., “Energy-aware Virtual Machine Migration for Cloud Computing-a Firefly Optimization Approach”, *Journal of Grid Computing*, vol. 14, no. 2, pp. 327-345, 2016.
- [9] Gao, Y., *et al.*, “A Multi-objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing”, *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1230-1242, 2013.
- [10] Karaboga, D., and Ozturk, C., “A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm”, *Applied Soft Computing*, vol. 11, no. 1, pp. 652-657, 2011.
- [11] Ferdous, M. H., *et al.*, “Virtual Machine Consolidation in Cloud Data Centers using ACO Metaheuristic”, In *European Conference on Parallel Processing (pp. 306-317)*, Springer, Cham, 2014.
- [12] Buyya, R., *et al.*, “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities”, In *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on (pp. 1-11). IEEE, June 2009*.
- [13] Beloglazov, *et al.*, “Energy-aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing”, *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768.
- [14] Voorsluys, *et al.*, “Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation”, In *IEEE International Conference on Cloud Computing (pp. 254-265), Springer, Berlin, Heidelberg, December 2009*.
- [15] Buyya, R., *et al.*, “Energy-efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges”, 2010, arXiv preprint arXiv:1006.0308.
- [16] Beloglazov, A., and Buyya, R., “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*, vol. 24, no. 13, pp. 1397-1420, 2012.
- [17] Theja, P. R., and Babu, S. K., “Evolutionary Computing Based on QoS Oriented Energy Efficient VM Consolidation Scheme for Large Scale Cloud Data Centers”, *Cybernetics and Information Technologies*, vol. 16, no. 2, pp. 97-112, 2016.
- [18] Shidik, G. F., *et al.*, “Evaluation of Selection Policy with Various Virtual Machine Instances in Dynamic VM Consolidation for Energy Efficient at Cloud Data Centers”, *JNW*, vol. 10, no. 7, pp. 397-406, 2015.
- [19] Namdev, S., *et al.*, “Improved Minimum Migration Time VM Selection Policy for Cloud Data Center”, *International Journal of Application or Innovation in Engineering and Management*, vol. 4, no. 4, pp. 157-60, 2015.
- [20] Mevada, A., *et al.*, “Enhanced Energy Efficient Virtual Machine Placement Policy for Load Balancing in Cloud Environment”, *Int J Cur Res Rev*, vol. 9, no. 6, p. 50, 2017.

BIOGRAPHIES OF AUTHORS



A. R. Mohazabiyeh IT engineer and received his B.S degree from Bushehr University in 2012 and a M.S degree in computer networks from the Islamic Azad University of Bushehr 2017.



Khosrow Amirizadeh received his B.S. degree in computer engineering from Shiraz University in 1990 and his M.S. degree in Artificial Intelligence from IAU- Researches and Sciences in 1998, Tehran, Iran and PhD degree from University Sains Malaysia (USM) in 2014. Currently he is working on Reinforcement learning and application in Neuroimaging tasks, Optimization problems, Intelligent control and decision making.