Journal of Computing and Security

Research Article, 2023, Volume 10, Number 2 (pp. 83–92)

http://www.jcomsec.org

A Multi-Dimensional Framework for Virtual Machine Consolidation

Sayed Ali Mossavi [a] Abbas Horri [b,*]

[a] *Department of Computer Science, Faculty of Computer Science, University of Kabul, Kabul, Afghanistan.*

[b] *Computer Engineering Department, Faculty of Engineering, Shahrekord University, Shahrekord, Iran.*

ARTICLE INFO.

Article history:
Received: 10 Nov 2022
Revised: 26 Jun 2023
Accepted: 02 Sep 2023
Published Online: 27 Jan 2024

Keywords:
Cloud Computing, Green Computing, Virtual Machine, Quality of Service, Energy Efficiency.

ABSTRACT

Cloud computing is a demand computing model that requires a large number of physical resources and provides services based on the request of each user. A large number of physical servers in data centers have high electrical energy consumption, which causes high operating costs, increases carbon dioxide (CO2) emission. The focus of this paper is on virtual machine consolidation to minimize power consumption, the number of VM migrations and, reducing service level agreement violation. In contrast to the existing works that use CPU utilization for the detection of host overload, a recent study has proposed Multiple Regression Host Overload Detection (MRHOD) and Hybrid Local Regression Host Overload Detection (HLRHOD) algorithms which take multiple factors (CPU, memory, and network bandwidth utilization) into consideration. This paper provides a framework that takes into account multiple factors: CPU, memory, and bandwidth utilization in three terms: host overload detection, VM placement, and service level agreement violation. First, in the host overload detection term, we provide a Separately Local Regression Host Overload Detection (SLRHOD) algorithm that considers CPU, memory, and bandwidth utilization, separately. Second, in terms of VM placement which is an NP-hard problem, the Power Aware Best Fit Decreasing (PABFD) algorithm with consideration of Dot-product (DP) heuristics was proposed. Third CPU and memory take into account the calculation of SLA violation in terms of SLA violation. To evaluate our framework in contrast to existing works, many experiments were performed. For each experiment, we evaluated and compared three objectives, namely energy consumption, service level agreement violation, and the number of VM migrations. Our experiment results show that the Separately Local Regression Host Overload Detection (SLRHOD) algorithm in terms of SLA violations reveals a significant improvement of 80%. On the other hand, the Separately Local Regression Host Overload Detection (SLRHOD) algorithm saves energy, up to 3%, compared to the HLRHOD and MRHOD algorithms. Our simulation results show that our proposed algorithm outperforms the existing algorithm and achieves improvements in energy consumption, service level agreement violation, and the number of VM migrations.

1 Introduction

Cloud computing is a rapidly growing paradigm and has revolutionized the communication technology (ICT) industry and information by considering many exciting features like on-demand computing resources, operational cost, elastic scaling elimination of up-front capital, and establishing a pay-as-you-go business model for information technology services and computing [1]. Nowadays many organizations and business owners instead of setting up data centers, have begun to use cloud services. Due to the increasing need of today's world to use these services, cloud providers have begun to create multiple large-size data centers. The extension of cloud computing has led to the construction of large data centers including thousands of computers around the world that consume high electrical energy. In 2012, energy consumption by data centers worldwide was 300 - 400 tera-watt per hour, about 2% of the global electricity usage, and it is estimated to triple by 2020, see Fig. 1 [2].

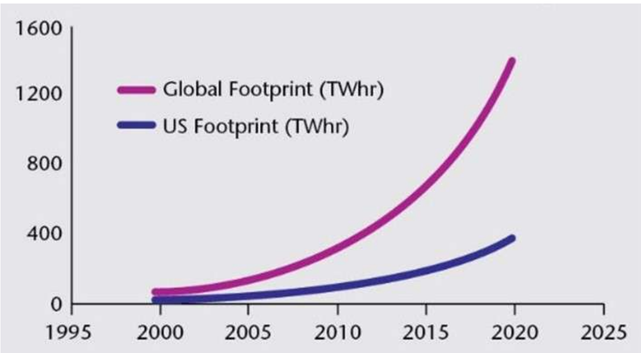

Figure 1. Projection of Data Centers' Electricity Uses [2].

High electrical energy consumption in data centers, results in high operating costs and carbon dioxide (CO2) emissions. Carbon dioxide (CO2) emissions cause serious harm to the environment. Hence, the reduction of energy consumption in data centers for both personal gain and harm reduction in the environment is changed to an important area. The quantity of computing resources and power inefficiency of hardware is not the only reason for high energy consumption but rather lies in the inefficient use of resources. One way to reduce energy consumption can be achieved by eliminating the inactive power consumption from computing resources and switching inactive nodes to low-powered modes (i.e., hibernation, sleep). According to the current resource requirements by using live VM migration [3] the VMs can be dynamically consolidated to the minimal number of physical servers. Host overload detection and VMs migration of most of the recent research is based on CPU utilization of physical servers. CPU-based models may be accurate and suitable for CPU-intensive applications, but will not be accurate for other types of applications like network, I/O, and memory-intensive applications [4]. On the other hand, a high computing application, and server applications usually utilizing cloud data centers, are sensitive to multiple factors (CPU, memory, and Network BW utilization).

Therefore, in this paper, we provide a framework that takes into account multiple factors: CPU, memory, and bandwidth utilization in three terms: host overload detection, VM placement, and service level agreement violation. We provide a Separately Local Regression Host Overload Detection (SLRHOD) algorithm in terms of the host overload detection term, Multiple-factor VM placement (MFVMP) algorithm is proposed in terms of VM placement, and terms of SLA violation calculation CPU and memory are taken into consideration. The rest of this paper is arranged as follows: Section 2 discusses related works. Section 3 discusses host overload detection. Section 4 explains VM placement. Section 5 describes the evaluation methodology. Section 6 explains the algorithms' comparative analysis using the random workload. Section 7 discusses performance evaluation. Experimental setup and experimental results are explained in Section 8 and Section 9. Finally, in Section 10 the discussion is on the conclusions and future works.

2 Literature Review

Dynamic virtual machine consolidation has an important role in reducing electrical energy consumption. Therefore, researchers trying to use this technique to solve problems optimally to provide energy consumption reduction and quality of service assurance. So in this section, we focus on the most relevant work.

Adaptive energy-aware algorithms have been proposed in [5] to maximize energy efficiency and minimize SLA violations in data centers. Unlike the existing methods, these energy-aware algorithms consider a variety of applications as well as CPUs and memory resources when establishing virtual machines. The proposed methods using real-world workloads, and extensive analysis has been done on more than a thou-

* Corresponding author.
Email addresses: alimosavi.cs@gmail.com (A. Mossavi), Horri@sku.ac.ir (A. Horri)
https://dx.doi.org/10.22108/JCS.2023.135300.1112
ISSN: 2322-4460

sand PlanetLab VMs. Experimental results show that compared to the existing energy-saving methods, the proposed approaches meantime maintaining low SLA violations can decrease power consumption in cloud data centers.

Reddy, V.D, et al. [6] proposed a modified discrete particle swarm optimization algorithm based on the characteristic particle swarm optimization for the initial placement of virtual machines and a novel virtual machine selection algorithm for optimizing the current allocation based on memory utilization, bandwidth utilization, and size of the virtual machine. By using simulation tools, the results show that the proposed method not only saves energy significantly compared to other methods but also minimizes the violation of service level agreements (SLA).

Zuo, Liyun, et al. [7] proposed the self-adaptive threshold based on the dynamic weighted load method. The proposed method divides the resource load into three states, including overflow, normal, and idle. It then migrates those overload resources to a balanced load and releases the idle resources whose idle times exceed a threshold to save energy. The results of the experiment show that the proposed method is more successful in maintaining energy, response time, and resource efficiency than other models and has been improved in these cases.

OpenStack Neat presented by Beloglazov, Anton [8] to provides a large framework for the dynamic consolidation of VMs. This project will be implemented in the form of separated services from the core OpenStack. This project will interact with the core OpenStack Services by using the public APIs of OpenStack services. OpenStack Neat is used as the Terracotta [9] codebase at the very primary stage. TerraCotta is a dynamic resource consolidation of OpenStack, which implements dynamic consolidation of resources, e.g. Virtual Machines (VMs) using live migration.

The authors in [10], have discussed the reduction of power consumption of physical servers in clouds. Their model calculates overall energy by adding up the energy consumption of both CPU and RAM. The power difference of a server before allocation and after allocation is taken into account while VMs are placed on the server, which has never been picked up for VM placement. To control SLA violations, the authors used various threshold mechanisms.

Khoshkholghi, Mohammad Ali, et al. [11] proposed a dynamic VM consolidation to improve the utilization of resources, such as CPU, RAM, and bandwidth in cloud data centers. In their study, the authors in their study proposed algorithms to reduce energy consumption and improve the performance of computing resources in overall data centers based on the SLA. Results show an improvement in energy costs as well as the performance of data centers.

Recently, in [12] the authors have proposed a multi-factor energy-efficient resource allocation model. The proposed model takes into account the joint power consumption of CPU and RAM while taking VM hosting decisions. In this work, the capacity of the server is considered along with the power consumption while selecting a server for the hosting of VMs.

Most of the above work relies on a single factor (CPU utilization) or three factors (CPU, RAM, and BW utilization) to detect overloaded host while the mentioned effective parameters are important in VM placement and SLA violation. Therefore, in this paper due to the mentioned problems, we take into account CPU, RAM, and network bandwidth parameters in three terms: host overload detection, VM placement, and calculating of SLA violation. In other words, in this paper, one of our new works that distinguishes our work from previous works is that we take into account CPU and RAM in SLA violation calculation (SLATAH and PDM) which any previous works have not been considered so far.

3 Host Overload Detection

Most of the recent works use only one factor (CPU utilization) for detecting the host overload and ignore the effective parameters such as memory and BW. Therefore, we tried to consider all three parameters (CPU, memory, and BW utilization) for host overload detection in the following algorithms by using multiple regression and local regression.

3.1 Multiple Regression

Multiple regression [13] is an extension of simple linear regression. The purpose of multiple regression is to predict the value of a variable based on the value of two or more other variables. The predicted variable is called the dependent variable. The variables that we are using to predict the value of the dependent variable are called the independent variables. Generally, multiple regression explains the relationship between predictor variables or multiple independent here (CPU, memory, and BW) and one criterion variable or dependent here (host utilization). The outcome of the multiple regression algorithm is a prediction of future host utilization. After the predicted host utilization a VM is selected to be migrated from the overloaded host. For this purpose, the Minimum Time Migration (MMT) [14] algorithm is used in all the following overload detection algorithms for the selection of virtual machines.

3.2 Multiple Regression Host Overload Detection(MRHOD)

The pseudo-code for MRHOD is presented in Algorithm 1. This algorithm proposed in [1] uses Geometric Relation (GR) which is suggested in [15]. Geometric Relation is a multi-parameter relationship that combines multiple parameters in one metric. The most critical factors to be considered for VMs are CPU, memory, and BW. On the other hand, the absolute values for those factor scores are not the desired parameters to be used. The utilization of the aforementioned factors relative to the maximum permissible utilization is more meaningful to make the factors dimensionless and representative of the host overload. The equation of this geometric relation used in [16] is shown in Eq. 1.

$$\text{HostUtilization} = \frac{W1}{1-\text{CPU}} \times \frac{W2}{1-\text{RAM}} \times \frac{W3}{1-\text{BW}} \tag{1}$$

Where: W_i: CPU, memory, and BW weight, CPU: CPU utilization, RAM: memory utilization, and BW: network utilization.

Algorithm 1 Multiple Local Regression Host Overload Detection (MLRHOD)

Input: CPU, RAM, and BW utilization history
Output: A decision whether the host is overloaded

1: **for** each host in host_list **do**
2:   **for** each sample of data **do**
3:     $X \leftarrow$ Multidimensional matrix {CPU, RAM, BW}
4:     $Y \leftarrow \frac{W1}{1-\text{CPU}} \times \frac{W2}{1-\text{RAM}} \times \frac{W3}{1-\text{BW}}$
5:     Use OLSMultipleLinearRegression
6:     Estimate the Multiple regression parameters
7:     Calculate the prediction utilization using multiple regression model
8:     **if** prediction utilization < 1 **then**
9:       Repeat steps 2 to 8
10:     **end if**
11:   **end for**
12: **end for**
13: **return** prediction utilization ≥ 1 and host is overloaded

MRHOD algorithm first calculated CPU, RAM, and BW utilization for each host, then gives two metrics as input to multiple regression. "The first one is a two-dimensional array named X with the size of rows equal to the length of the Data and column size equal to the number of independent parameters which are three in this algorithm (CPU utilization, RAM utilization, and BW utilization) [1] ". "The second input is one dimension array named Y which includes a sample of data of host utilization calculated from Equation (1) [1]". Besides that, the OLSMultipleRegression(Ordinary Least Square Multiple Regression's) function [17, 18] to calculate the regression coefficients (parameters) of the multiple regression model. After the calculation of regression coefficients the predictiedhostutilization equation formed as shown in Eq. 2:

$$\begin{aligned}\text{predicted utilization} &= b_0 + (b_1 \cdot \text{CPU utilization}) \\ &+ (b_2 \cdot \text{Ram utilization}) + (b_3 \cdot \text{Bw utilization}) + \ldots\end{aligned} \tag{2}$$

The triggering point of the algorithm is that if the predictedUtilization is greater than or equal to 1 then the host is considered to be overloaded so we select a VM to be migrated from the overloaded host.

3.3 Local Regression

Local regression is used to model the relationship between a dependent variable and an independent variable. Here the purpose of a dependent variable is one or a combination of three parameters (CPU, memory, and BW), and host utilization is an independent variable. In the end, the outcome of the local regression algorithm is a prediction of future host utilization.

3.4 Hybrid Local Regression Host Overload Detection Method(HLRHOD)

The proposed HLRHOD algorithm in [1] uses geometric relations and the formula developed in [15] by using a metric that captures the combined CPU, memory, and BW load of virtual and physical servers to calculate the host utilization based on hybrid factors. They use the Local Regression (LR) host overload detection algorithm proposed by Cleveland [14] after calculating the host utilization from the above formula. However, if the value of the predicted metric (predictedUtilization) is greater than or equal to 1, the host utilization is considered as an overloaded host, which in this case is a virtual machine selected to migrate from an overloaded host to underloading host.

3.5 Separately Local Regression Host Overload Detection (SLRHOD)

One of our proposed algorithms is host overload detection which uses local regression. We use local regression to detect overloaded host in this algorithm, which is based on the method proposed by Cleveland [14], but with the difference, that along with the CPU, memory and network parameters are considered also separately. In simple words, we can say that we use OR logic relation between CPU, memory, and BW for host overload detection, it means that in our algorithm any of the parameters (CPU, memory, and BW) of a host does not have enough capacity is considered to be an overloaded host. Finally, by substituting the CPU, memory, and BW utilization values into the local regression equation, we obtain a host utilization prediction. In this case, if the value of the predictedUtilization metric of any of the parameters (CPU, memory, and BW) is equal to or greater than 1, the host is considered overload, in that case, a virtual machine selected for migration from overload host to underloading host.

4 VM Placement Algorithm

VM Placement (VMP) problem and selecting a new host for them can be supposed as a bin packing problem, here items are considered as VMs and bins as PMs/hosts. Since the bin packing problem is an NP-Hard problem, Beloglazov and Buyya proposed the PABFD algorithm [19]. The pseudo-code for this algorithm is presented in Algorithm 2.

Algorithm 2 Power Aware Best Fit Decreasing (PABFD)

Input: hostList, vmList
Output: allocation of VMs

```
Sort VMs based on utilization in decreasing order
for each vm in vmList do
    minPower ← MAX
    allocateHost ← NULL
    for each host in hostList do
        if (host has enough resources for vm) then
            Power ← estimatePower(host, vm)
            if (Power < minPower) then
                allocateHost ← host
                minPower ← Power
            end if
        end if
    end for
    if (allocateHost ≠ NULL) then
        allocate vm to allocateHost
    end if
end for
return new VMs placement
```

This algorithm has a very good effect on the amount of energy consumption but isn't still a complete algorithm. Therefore PABFD algorithm is not considered an important allocation, or service level agreement violation because the main focus of this algorithm is on reducing energy consumption.

Now, in addition to the Buyya and Beloglazov method, in this section what we will present is the selection of a new destination for virtual machines using the PABFD algorithm based on the Dot-product (DP) heuristic method, which in addition to CPU, memory factor is also considered. This algorithm is called the Multi-factor VM Placement Algorithm (MFVMP). The pseudo-code for this algorithm is presented in Algorithm 3. In this algorithm, the DP heuristic method works in such a way that not only considers the demand of a virtual machine but also considers the remaining bin capacity. This algorithm includes two stages, sorting and placement of virtual machines. In the first stage, all virtual machines are sorted based on CPU utilization in descending order. In the second stage, this algorithm for VM placement uses the Dot-product (DP) heuristic method. One of the most important features of this heuristic is that it does not ignore the remaining capacity of the bin and places virtual machines over the bin so that both dimensions, CPU, and memory are filled at the same time. This feature of this heuristic makes us see a better result in the degree of service level agreement violation. As the results will show in the evaluation section replacing the PABFD algorithm with the proposed MFVMP algorithm will significantly reduce service level agreement violation.

Algorithm 3 Multi-factor VM Placement (MFVMP)

Input: hostList, vmList
Output: allocation of VMs

```
Sort VMs based on utilization in decreasing order
for each vm in vmList do
    minDotProduct ← MAX
    allocateHost ← NULL
    for each host in hostList do
        if (host has enough resources for vm) then
            dtProduct ← dotProduct(host, vm)
            if (dtProduct < minDotProduct) then
                allocateHost ← host
                minDotProduct ← dtProduct
            end if
        end if
    end for
    if (allocateHost ≠ NULL) then
        allocate vm to allocateHost
    end if
end for
return new VMs placement
```

5 Evaluation Methodology

Cloud computing environments provide users with an infinite view of computing resources. Since we need to evaluate the algorithms presented in a large-scale data center. However, it is difficult to do this in the real world and on real infrastructure. Therefore, to evaluate our proposed algorithms, simulation has been used. There are many advantages of simulation tools: no capital cost, providing perfect results, by using such tools helps to change inputs and other factors also in an easier way which results in better and efficient outputs. Learning simulation tools is easy, when working with such simulation tools a user needs to have only programming abilities [20]. Therefore, we choose the CloudSim 3.0.3 toolkit as our simulation platform.

Cloudsim is a platform for simulating and modeling cloud computing services and infrastructures. It was launched in 2009 by the GRIDS Laboratory at the University of Melbourne. Cloudsim is an open-source project that was developed in JAVA. The platform not only provides the infrastructure mode of cloud computing but also provides an interface for resource management in a cloud environment [21]. This simulation platform has the following core classes: VMProvisioner, DataCenterBroker, Cloudlet, DataCenter, Host, VirtualMachine, VMCharacteristics, VMScheduler, and VMMAllocationPolicy. Recently some studies developed such a platform by enhancing CloudSim [22].

6 Workload

In the simulation study, experiments using real workload data help to make precise conclusions. The virtual machine resource utilization was simulated based on the data provided in the CoMon project, a monitoring infrastructure for PlanetLab. This dataset contains the utilization of, CPU, bandwidth, and memory of 1000 hosts that are placed around the world. The data was collected during the period between the 3rd of March to the 20th of April 2011. In this paper, in terms of host overloaded detection, we provide a Separately Local Regression Host Overload Detection (SLRHOD) algorithm, Multiple-factor VM placement (MFVMP) algorithm is proposed in terms of VM placement and terms of SLA violation calculation CPU and memory take into consideration.

7 Performance Metrics

The performance metrics used to compare and evaluate our proposed algorithms are summarized as follows:

- **Total energy consumption (E):** The total energy is the amount of energy that is consumed by the physical resources of a data center. Whatever this parameter is lower to the purpose of executing workload, it is more desirable.
- **The number of VM migrations:** The VMs selected for migration in dynamic VM consolidation when overloaded or under-loaded hosts are found. Therefore, this metric is an important metric that reflects the time needed to migrate VMs from the overloaded host to other under-loaded hosts.
- **SLATAH:** One of the differences between our work and previous work is that we consider the effective memory parameter in addition to the CPU parameter for calculating the service level agreement violation. In the first step, the SLATAH of both parameters CPU and memory calculated, and then the maximum of these parameters is considered a service level agreement violation. SLATAH is the percentage of the time, during which active hosts have experienced the CPU or RAM utilization of 100%. SLA violation Time per Active Host (SLATAH) as shown in Eq. (3):

$$SLATAH = \frac{1}{M}\sum_{i=1}^{m}\frac{T_{si}}{T_{ai}} \tag{3}$$

  Where M is the number of hosts, T_{si} is the time when full utilization has been experienced by server i leading to an SLA violation, and T_{ai} is the total time that server i stays active.
- **PDM:** Performance Degradation due to Migration (PDM) is the overall performance degradation caused by VMs due to migrations. In this metric, in addition to CPU, also the effective memory parameter is considered in the calculation of service level agreement violation which is one of our different works compared to the previous work. In this metric, the first PDM of both parameters is calculated and then the maximum of them is considered as the service level agreement violation. Performance Degradation due to Migrations (PDM) as shown in Eq. (4):

$$PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{dj}}{C_{rj}} \tag{4}$$

  Where M is the number of VMs, C_{dj} is the estimate of the performance degradation of the VM j caused by migrations, and C_{rj} is the total CPU capacity requested by the VM j during its lifetime.

In this paper, in terms of host overloaded detection, we provide a Separately Local Regression Host Overload Detection (SLRHOD) algorithm, Multiple-factor VM placement (MFVMP) algorithm is proposed in terms of VM placement and terms of SLA violation calculation CPU and memory take into consideration.

- **SLA violation (SLAv):** SLA violation occurs when the promised QoS cannot be delivered to the VM [22, 23]. (SLAV) metric is the product of (3) and (4), defined as follows:

$$SLAV = SLATAH * PDM \qquad (5)$$

8 Experimental Setup

In this section, various experiments will be explained to evaluate and compare the proposed method with other methods. These experiments were performed using the CloudSim library with the following parameters:

- **Simulation Time:** This is the simulation run time, which is defined by using the SIMULATION-LIMIT parameter. The value of this parameter in all experiments is equal to 24 hours which 86400 seconds is considered.
- **Data Center:** To perform experiments, a data center is defined that includes several servers.
- **Physical Server:** Two different types of servers are defined in the data center. The number of data center servers is defined as a variable that can change in
- **Workload:** The workload of applications will be generated randomly by Cloudlet. The length cloudlet of cloud 25000 * SIMULATION-LIMIT is considered. During simulation runtime, the amount of requested processing power changes by the cloudlet randomly.
- **Scheduling Interval:** The operation optimization will be performed every 5 minutes (300 seconds) on the current allocation of virtual machines.

9 Experimental Results

Our data center consists of 50 physical servers and 50 virtual machines in this experiment, which was repeated 10 times. In this experiment, we compared overload host (HLRHOD-MMT, MRHOD-MMT, and SLRHOD -MMT) algorithms based on two VM placements (MFVMP and PABFD) algorithms. Their diagrams are based on the different experiments. The specifications of the servers in Table 1 and the status of their power consumption are given in Table 2 as follows:

Table 1. Power Consumption in Watts at Different Status.

Server	Power Consumption Based on CPU Utilization					
	0%	20%	40%	60%	80%	100%
HP ProLiant G4	86	92.6	99	106	112	11
HP ProLiant G5	93.7	101	110	121	129	135

Table 2. PM Configuration.

RAM(GB)	Frequency(MHz)	Cores	CPU Model	Server
HP ProLiant G4	Intel Xeon 3040	2	1860	4
HP ProLiant G5	Intel Xeon 3075	2	2660	4

performance metrics mentioned above and displayed in the following figures.

Fig 2 Shows energy consumption evaluation, which our proposed SLRHOD outperforms the HLRHOD and MRHOD algorithms by up to 3% because SLRHOD can switch more underloaded hosts to the sleep mode, which lowers power consumption compared to the idle state, leading to more energy savings by hosts.

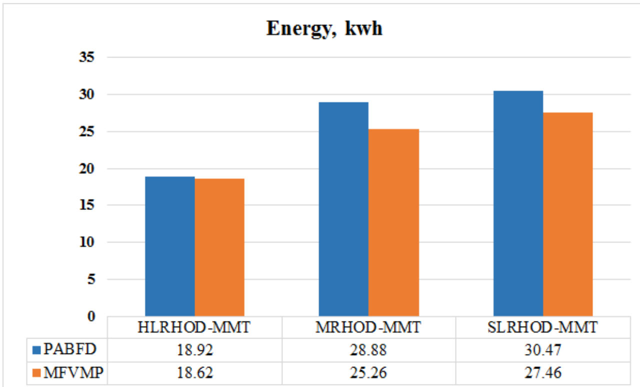

Figure 2. Energy Consumption.

Fig 3 The result indicates that SLRHOD compared to the HLRHOD and MRHOD algorithms, SLRHOD dramatically decreases the number of VM migrations. On the other hand, in the VM migration term, a mechanism is preferred that requires fewer migrations to consolidate the VMs because live VM migration imposes an overhead on the system; hence, cloud administrators set a limiting number of migrations depending on the acceptable VM migration overhead.

Fig 4 Shows the PDM metric result. The graph shows that the PDM of the SLRHOD algorithm compared to HLRHOD and MRHOD has been reduced by 7.5% and 50%, respectively.

Fig 5 Shows that SLRHOD outperforms SLATAH rather than HLRHOD and MRHOD algorithms. The results indicate that SLATAH of SLRHOD algorithm

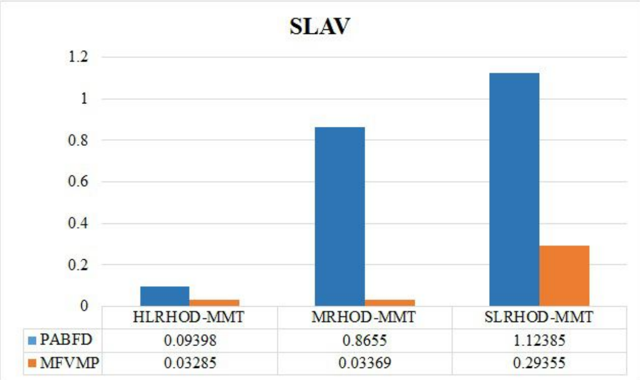

Figure 3. SLA Violation.

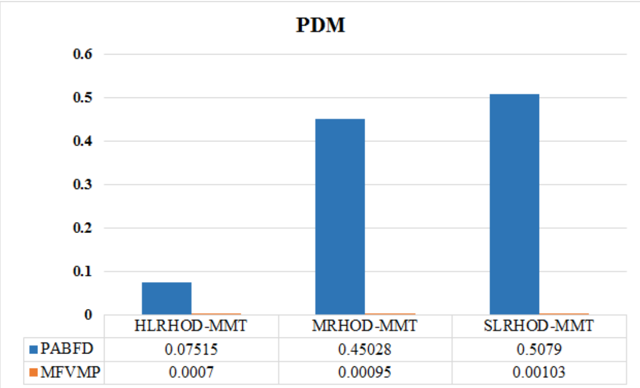

Figure 4. PDM.

compared to HLRHOD and MRHOD have been reduced by 1.5% and 3%, respectively.

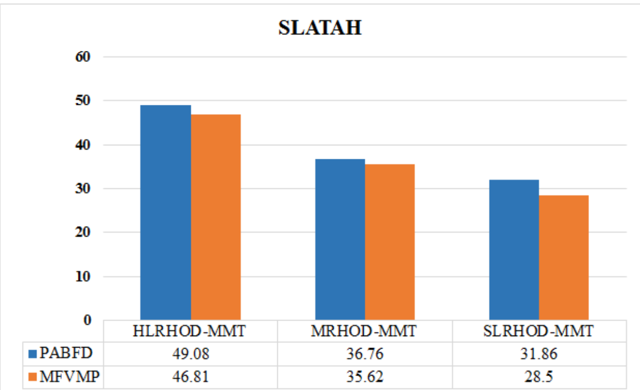

Figure 5. SLATAH.

Fig 6 Demonstrates that SLRHOD dramatically reduces SLA violations compared to HLRHOD and MRHOD algorithms. SLRHOD algorithm than HLRHOD and MRHOD algorithms have been reduced by 6% and 83%, respectively, in terms of SLAV.

The achieved results by SLRHOD in terms of SLA violations reveal a significant improvement of 80%. On the other hand, SLRHOD saves energy, up to 3%,

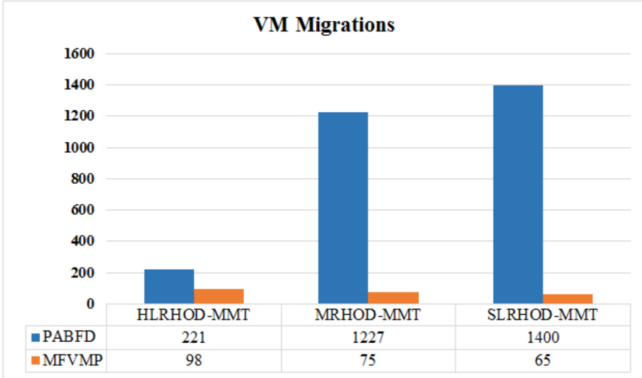

Figure 6. VM Migration.

compared to the HLRHOD and MRHOD algorithms. Finally, it can be said that our proposed framework can be adapted to a real environment because we consider not only the CPU, but also many resources RAM, and BW in three dimensions; host overload detection, VM placement, and SLA violation.

10 Conclusions and Future Works

In our suggested framework, we take into account multi-factor in all dimensions (overload detection, VM placement, and SAL violation); whereas, in the recent studies, some of them have taken multi-factor resources into account which are chosen in terms of host overload detection or VM placement, but not in all terms. Therefore, the implementation of our suggested framework can operate in the real environment of a data center, because all the effective parameters in all dimensions are considered. The comparison of our algorithms with benchmark algorithms shows that our framework outperforms the benchmark algorithms in all terms; in terms of energy consumption, VM migration, PDM, SLATAH, and SLAV. Finally, our future work is that we will consider the addition of CPU, RAM, and BW as the storage parameter in calculating SAL violation and VM placement.

References

[1] A. Abdelsamea, A. A. El-Moursy, E. E. Hemayed, and H. Eldeeb. Virtual machine consolidation enhancement using hybrid regression algorithms. *Egyptian Informatics Journal*, 18(3):161–170, 2017. doi:https://doi.org/10.1016/j.eij.2016.12.002.

[2] S. Ismaeel, R. Karim, and A. Miri. Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres. *Journal of Cloud Computing*, 7(1):10, 2018. doi:https://doi.org/10.1186/s13677-018-0111-x.

[3] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 273–286, 2005. doi:10.1145/1064413.1064440.

[4] G. Dhiman, K. Mihic, and T. Rosing. A system for online power prediction in virtualized environments using gaussian mixture models. In *Proceedings of the 47th Design Automation Conference*, pages 807–812, 2010. doi:https://doi.org/10.1145/1837274.1837478.

[5] Z. Zhou, J. Abawajy, M. Chowdhury, Z. Hu, K. Li, H. Cheng, A. A. Ale-laiwi, and F. Li. Minimizing sla violation and power consumption in cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*, 86:836–850, 2018. doi:https://doi.org/10.1016/j.future.2017.07.048.

[6] V. D. Reddy, G. Gangadharan, and G. S. V. Rao. Energy-aware virtual machine allocation and selection in cloud data centers. *Soft Computing*, 23(6):1917–1932, 2019. doi:https://doi.org/10.1007/s00500-017-2905-z.

[7] L. Zuo, L. Shu, S. Dong, C. Zhu, and Z. Zhou. Dynamically weighted load evaluation method based on self-adaptive threshold in cloud computing. *Mobile Networks and Applications*, 22(1): 4–18, 2017. doi:https://doi.org/10.1007/s11036-016-0679-7.

[8] A. Beloglazov and R. Buyya. Openstack neat: A framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds. *Concurrency and Computation: Practice and Experience*, 27(5):1310–1333, 2015. doi:https://doi.org/10.1002/cpe.3314.

[9] Terracotta project, 2020. URL `https://terracotta.readthedocs.io/en/latest/project_info`.

[10] P. H. Castro, V. L. Barreto, S. L. Correa, L. Z. Granville, and K. V. Cardoso. A joint cpu-ram energy efficient and sla-compliant approach for cloud data centers. *Computer Networks*, 94:1–13, 2016. doi:https://doi.org/10.1016/j.comnet.2015.11.026.

[11] M. A. Khoshkholghi, M. N. Derahman, A. Abdullah, S. Subramaniam, and M. Othman. Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers. *IEEE Access*, 5:10709–10722, 2017. doi:10.1109/ACCESS.2017.2711043.

[12] B. Gul, I. A. Khan, S. Mustafa, and O. Khalid et al. Cpu-ram-based energy-efficient resource allocation in clouds. *The Journal of Supercomputing*, 75(11):7606–7624, 2019. doi:https://doi.org/10.1007/s11227-019-02969-5.

[13] M. A. H. Monil and R. M. Rahman. Vm consolidation approach based on heuristics, fuzzy logic, and migration control. *Journal of Cloud Computing*, 5(1):8, 2016. doi:https://doi.org/10.1186/s13677-016-0059-7.

[14] A. Beloglazov 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*, 24(13):1397–1420, 2012. doi:https://doi.org/10.1002/cpe.1867.

[15] J. Chen, W. Liu, and J. Song. Network performance aware virtual machine migration in data centers. In *CLOUD COMPUTING*, pages 65–71, 2012.

[16] T. Anandharajan, D. Bhargavan, and M. Bhagyaveni. Vm consolidation techniques in cloud datacenter. *Journal of Theoretical & Applied Information Technology*, 53(2), 2013.

[17] W. Tian, Y. Zhao, Y. Zhong, M. Xu, and C. Jing. A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters. In *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, pages 311–315, 2011. doi:10.1109/CCIS.2011.6045081.

[18] M. Tang and S. Pan. A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. *Neural Processing Letters*, 41(2):211–221, 2015. doi:https://doi.org/10.1007/s11063-014-9339-8.

[19] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5):755–768, 2012. doi:https://doi.org/10.1016/j.future.2011.04.017.

[20] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation

of resource provisioning algorithms. *Software: Practice and experience*, 41(1):23–50, 2011. doi:https://doi.org/10.1002/spe.995.

[21] F. F. Moges and S. L. Abebe. Energy-aware vm placement algorithms for the openstack neat consolidation framework. *Journal of Cloud Computing*, 8(1):2, 2019. doi:https://doi.org/10.1186/s13677-019-0126-y.

[22] S. Saravanan, V. Venkatachalam, and S. T. Malligai. Optimization of sla violation in cloud computing using artificial bee colony. *Int. J. Adv. Eng*, 1(3):410–414, 2015.

[23] G. Han, W. Que, G. Jia, and L. Shu. An efficient virtual machine consolidation scheme for multimedia cloud computing. *Sensors*, 16(2):246, 2016. doi:https://doi.org/10.3390/s16020246.

Sayed Ali Mossavi received a B.Sc. degree in computer engineering from Kabul University, Kabul, Afghanistan, in 2015, and an M.Sc. degree in computer engineering from the University of Shahrkord, Shahrkord, Iran, in 2020, respectively. He is an Assistant Professor at the Department of Computer Science, Kabul University, Kabul, Afghanistan. His research interests include cloud computing, fog computing, green computing, Artificial Intelligence, and machine learning.

Abbas Horri received the M.Sc. and Ph.D. degrees in computer engineering from Shiraz University, Iran, in 2009 and 2014, respectively. He was the CIO (chief information officer) of Shahrekord University during 2015-2020. He is an Assistant Professor at the Department of Computer Engineering, Shahrekord University, Shahrekord, Iran. He has a pivotal role in establishing the Center for High-Performance Computing at Shahrekord University. Dr. Abbas Horri is also the founder of the Network Laboratory. His research interests include Grid Computing, Cloud Computing, Fog Computing, and Edge Computing.