

A NOVEL LOAD BALANCING APPROACH TOWARDS CLOUD COMPUTING BY REDUCING SLA DEGRADATION AND ENERGY CONSUMPTION

N.Nirmala Devi

Assistant Professor, Department of Computer Science, Auxilium College,
Vellore, India.
nirmalamca83@gmail.com

S.Vengatesh Kumar

Associate Professor, Department of Computer Applications (PG),
Dr.SNS Rajalakshmi College of Arts and Science,
Coimbatore, India
gowthamvenky@gmail.com

Abstract

The advent development of cloud computing is an on-demand adaptive technology for many IT organizations due to its dynamic scalability and virtualization resources as a global service. There are numerous energy-conscious approaches in existence that attempt to minimize energy consumption and Service Level Agreements (SLAs) degradation of the host. In this paper, we proposed a novel load balancing approach that identifies overloaded or underloaded hosts, and selects the VMs (Virtual Machines) for allocation to the host based on the predicted load. We consider CPU utilization parameter for evaluating host loading which is far enough to compute all the characteristics of Physical Machines (PM) provided to the Linear Adam Algorithm for Overloaded detection followed by Interquartile Range (IQR) method for underloaded detection. The Power-Aware Best Fit Decreasing (PABFD) algorithm sorts the VM in descending order based on CPU utilization. Then those filtered VM is allocated to the corresponding host thereby reducing the energy consumption and SLAs degradation. We evaluate the proposed algorithm through CloudSim-3.0.3 simulation and random datasets with different workloads on a real Planet Lab. After the evaluation, the simulation result of the proposed method indicates a significant performance by reducing the metric parameters such as SLA degradation, number of VM migration, the total number of host shutdowns, and reduce energy consumption.

Keywords: CPU utilization, SLA, VM, PM, Adam algorithm, Robust Static Interquartile Range, MMT, CloudSim.

1. Introduction

Cloud computing has shown immense development in IT infrastructure wherein it utilizes the dissipate resources benefits and provides computing services to end-users. It consists of interconnected homogenous and heterogeneous physical servers which offer on-demand services to the client/users based on the pay-as usage model in a minimal management effort. Therefore, it is requisite to manage the data center effectively due to the large set of computing resources such as CPU, RAM, storage, and bandwidth which consume enormous amounts of energy. Cooling equipment is required to maintain the system's stability and flexibility in cloud computing [1]. Hence, the high-level energy consumption leads to increased costs and CO₂ emissions, which induces the number of host shutdowns to maintain the adaptive strategies of SLA (Service Level Agreement) contract among the users and cloud providers. Through virtualization technology, cloud computing resources can be utilized in an optimized manner where it performs the live migration of VM. In the model cloud data center, the VM migrated takes place among the host during the overloaded/underloaded states, reducing the number of hosts and enabling the idle host to power-saving mode. However, the live migration may cause performance degradation and SLAs degradations, which has a massive impact on the QoS due to CPU, RAM, and bandwidth utilization. Therefore, it is crucial to decrease the energy consumption and SLAs cost, which results in NP-hard optimization problems. VM

consolidation is a productive technique to manage cloud computing resources that focuses on packing numerous VMs into a single host depending on the computing resources. It is divided into static and dynamic VM consolidation, moreover, dynamic VM consolidation is utilized, which allocates the VMs to suitable hosts. Sometimes, packing many VMs into a single host can cause poor QoS and affect the stability of the data center. Hence, it requires an effective dynamic VM consolidation which makes an intelligent decision to predict the overloaded /underloaded host and select the selective VM for allocating to the host. The primary motivation of this paper is to design a dynamic VM consolidation method based on load balancing among the host and effectively achieve lower energy consumption, reduce SLAs degradation with an efficient host shutdown, and low time complexity. The proposed approach comprises three modules that improve the efficient utilization of cloud computing resources. Firstly, Linear Adam [2] progresses the host overload detection, which is a versatile algorithm that provides an optimal solution to a high-dimensional machine learning problem. It evaluates the historical data of CPU utilization and determines the loading level. The evaluation results dynamically alter based on the historical workload of the CPU. Secondly, the host underload detection is carried out by Interquartile Range (IQR) method. Thirdly the VM selection is carried out by Minimum Migration Time (MMT) policy which selects the VM with minimum migration time to transact among source host and target host. The experimental result showed that the proposed load balancing approach uses real PlanetLab and random datasets with different workloads for analysis, enhancing the performance without affecting the response time of computing resources.

The contribution of the proposed methods is summarized below,

- (1) This paper devised a dynamic load balancing approach among the heterogeneous host and VMs to minimize the energy consumption, SLAs Degradation, and several host shutdowns.
- (2) The CPU utilization is minimized and reduces the computation time.
- (3) Linear Adam detects the overloaded host, which has the potential to solve the high-dimensional problems of machine learning.
- (4) The Robust Static Interquartile Range (IQR) method determines the lower threshold of CPU utilization based on requested MIPS (Million Instructions Per Second).
- (5) An efficient VM selection policy is carried out by MMT.
- (6) Performing extensive experiments using Real PlanetLab with a comprehensive VMs data collection and random datasets with different workloads successfully confirmed that the proposed method will be a promising solution for enhancing resource management among the overloaded/ underloaded hosts.

2. Related Work

In recent years, there are many approaches have been proposed on resolving the overloaded and underloaded host detection in cloud computing which minimizes the energy consumption of PM's and SLA degradation thereby providing an optimal resource to the host. In this section, we focused on the most relevant work on ESV and Load balancing concepts which will be highly supportive for refining our proposed model.

2.1. Energy and SLA Degradation (ESV)

Mapetu et al. [3] proposed a paper on reducing the trade-off between the SLA degradation and energy consumption reducing the host shutdowns and time complexity. Therefore, the Pearson correlation coefficient-based method is utilized to minimize the cost of SLA degradation. He tried to minimize the VM migration through mean deviation which is the product of imbalance degree and MAD. Even though, it theoretically provides a significant improvement in ESV that is not yet implemented in the real world. Mandal et al. [4], proposed better services to the cloud users through a power-aware policy that choose the specific VM that has less power consumption for migration. Still, the power-aware policy required some improvement in VM selection. Ibrahim et al. introduce a way to minimize the cloud data center's energy consumption through the Efficient Adaptive Migration Algorithm (EAMA). But it has SLA degradation conflicts which vigorously affect the resource allocation to the host. Kawsar et al. [5] presented a paper on the power consumption of the data center and utilized the regression-based algorithm to predict the appropriate host for VM migration with high CPU utilization.

2.2. Load Balancing Approach

Alhammadi et al. [6] analyzed the overloaded detection through the multiple regression method. They mainly investigate CPU, RAM, and BW utilization where require some improvement in VM consolidation. Yadav et al. [8] proposed an algorithm, namely GradCent (Stochastic Gradient Descent), which evaluates the CPU workload using an upper threshold value. It is applicable for predicting the overloaded hosts only. Moursy et al. [9] presented a paper on detecting host overload in cloud computing. The CPU utilization is evaluated through Multi-Dimensional Regression Host Utilization (MDRHU) algorithms via Euclidean distance, and it improves 12% of the energy metric in Overloaded. Masdari et al. [10] preview the VM migration techniques for cloud datacenters. Some methods approach CloudSim as a simulation environment, during the analysis, such as Markov-based, HMM-based, and Queuing Model-based predictive VM migration. He concluded that accurate predictive algorithms are needed for determining the host during overload/under loads conditions. Priyanka et al. [11] predict the host utilization of cloud computing through Support Vector Regression (SVR). SVR method is evaluated under the Alibaba dataset. They are planning to implement the work on PlanetLab workload trace in the future using CloudSim simulation. Kulkarni et al. [12] introduced a context-adaptive self-managing approach that dynamically assigns the VM to the host based on the physical characteristic of the machine. But it provides conflict results in SLA degradation. Nirmala et al. [15], describes the different type of hypervisor and virtualization technique utilized in the cloud computing environment. Mandal et al. [16] tried to evaluate the future resource utilization through a Linear Regression algorithm, but it fitted for a simple model and high complexity. Due to the performance degrades, it is not suitable for predicting resource utilization, and it also depends upon the linear relationship of data. Jararweh et al. [17] proposed a dynamic approach for resource utilization using Logistic Regression and Median Adaptive Deviation (MAD), it is widely suits for static environments, and the energy consumption is high. Among numerous studies, the research paper [16][17] provides a wide knowledge on resource utilization of cloud computing which supports our novel load balancing approach enhancement. Among all PMs, parameter CPU usage plays a vital role and provides enough data to estimate the energy consumption of PMs. Therefore, the CPU usage of PMs is acted as a key point for evaluating the number of migrations, host shutdowns, energy consumption, and SLA degradation in cloud computing.

3. Proposed System

This section provides a brief introduction to the Linear Adam and Interquartile range method. We then explain the proposed novel load balancing approach in cloud computing. The block diagram of the proposed approach is depicted in Fig. 1.

3.1. Linear Adam Algorithm

Linear Adam (adaptive moment estimation) algorithm is the architect on a stochastic gradient-based optimization approach where it estimates the lower-order moment values. The learning rate of the parameters is maintained as unfolds which is computationally efficient and utilized less memory, providing a solution for high dimensional machine learning problems. The large model training time is shortened and becomes a viable solution for optimization results. It drastically reduces the iteration of training and provides an accurate prediction [13]. It's also known as the straightforward approach and is well with large amounts of data. It requires little tuning during hyper-parameter interpretation. The product of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp) generate the Linear Adam approach. AdaGrad solves the sparse gradient problems maintaining the learning rate per parameter. RMSProp also maintains those per-parameter learning rates to adapt with the gradient weight value. Adam utilized the advantage of both AdaGrad and RMSProp, also make use of the second moment of the gradient. In the deep learning field, Adam provides good and fast results in data analysis. Compare to other optimization algorithms, Adam utilized the sparse feature for convergence thus it provides an optimal solution for the cloud environment.

3.2. Interquartile Range (IQR) Method

The interquartile range is a measure of statistical dispersion which lies between the upper and lower quartiles denoted as $IQR = QU3 - QU1$. It divides the data and measures the variability into quartiles where the data move under four equal parts. These data are called first QU1, second QU2, and third QU3 quartiles. IQR has a interquartile range of 25% based on that total range is evaluated. The Median Absolute Deviation (MAD) is a measure of half IQR range in a symmetric distribution. IQR is mostly used to estimate the outliers of input data and sort the data in increasing order.

3.3. Load Balancing Approach

The Load Balancing Approach problem is divided into four parts: i) Host Overloaded detection, ii) Host under loading detection, iii) VM Selection, iv) VM Allocation, and v) VM Migration. We discussed details in this section.

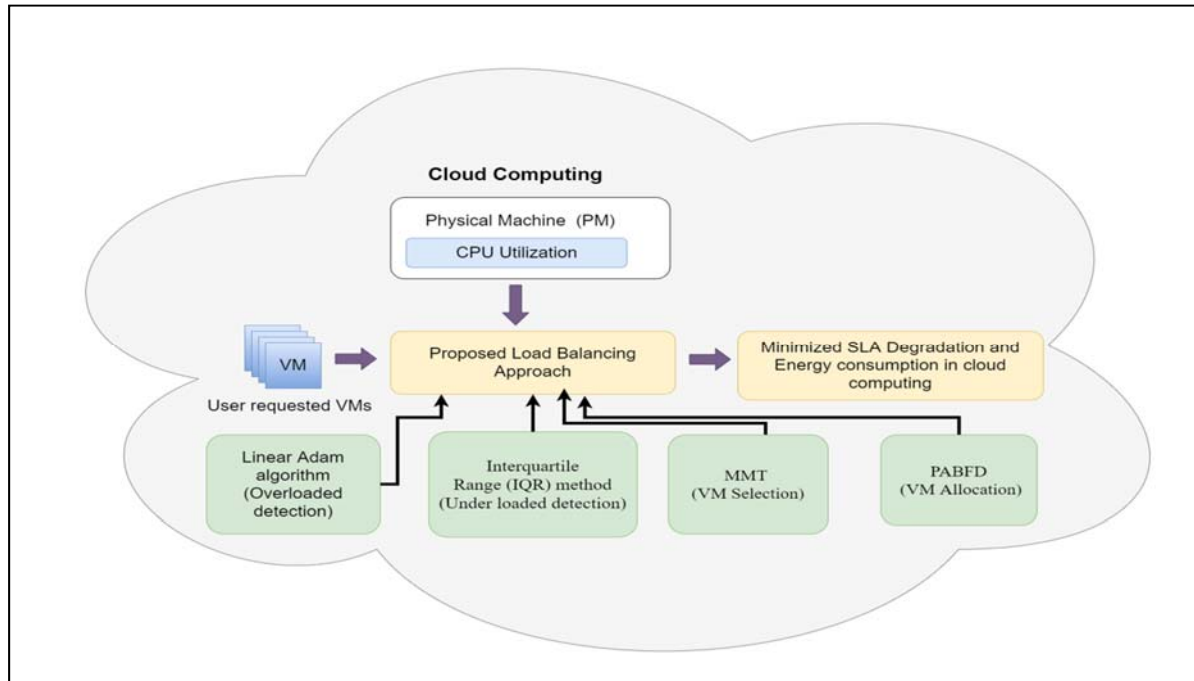


Fig. 1. Block Diagram of proposed Load Balancing Approach

3.3.1. Host Overloaded Detection

The proposed Linear Adam algorithm is utilized to detect the overloaded host, the pseudo-code of Host Overloaded algorithm is explained below,

ALGORITHM: ADM_HOST_OVERLOADED_DETECTION

Input: Host,

Output: isHostOverloaded

Default parameters:

$\alpha \leftarrow 0.001, \beta_1 \leftarrow 0.9, \beta_2 \leftarrow 0.999, \epsilon \leftarrow 10^{-8}, \text{MaxIteration} \leftarrow 1000$

$\theta \leftarrow ()$

$X \leftarrow [], Y \leftarrow []$

$\text{CpuUtilHistory} = \text{host.getUtilizationHistory}()$

If $\text{CpuUtilHistory.Length} < 12$ **then**

return $\text{host.getTotalRequestMips}() > 0.90 * \text{host.getTotalMips}()$

Else

$\text{CpuUtilHistory}[] \leftarrow \text{Reversing}(\text{CpuUtilHistory}[])$

For each $i = 1$ **to** m **do**

$X_i \leftarrow \text{CpuUtilHistory}(i)$

$Y_i \leftarrow \text{CpuUtilHistory}(i + 1)$

End

For t **in** MaxIteration

$g_t \leftarrow \nabla f_t(\theta_{t-1})$

$$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

$$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$

$$\widehat{m}_t \leftarrow \frac{m_t}{(1 - \beta_1^t)}$$

$$\widehat{v}_t \leftarrow \frac{v_t}{(1 - \beta_2^t)}$$

$$\theta_t \leftarrow \theta_{t-1} - \frac{\alpha \cdot \widehat{m}_t}{(\sqrt{\widehat{v}_t} + \epsilon)}$$

For End
Predicted $\leftarrow \theta_1 + \theta_2$ (*Current Host utilization (host)*)
 $Y \leftarrow \rho \times \text{Predicted}$
If $Y > P_{\text{threshold}}$ **then**
 return True
 Else
 return False
End
End If

$\alpha \leftarrow$ Learning parameter at each step (default is 0.1 and will be updated by the optimizer)
 $\beta_1 \leftarrow$ The first decaying average with proposed default value of 0.9 (deep learning purposes)
 $\beta_2 \leftarrow$ The second decaying average with proposed default value of 0.999 (deep learning purposes)
 $\epsilon \leftarrow$ A variable for numerical stability during the division
 $\text{MaxIteration} \leftarrow$ The number of Adam round we want to do before stopping the optimization
 $\rho = 0.5$ Adam's safety parameters
HostCpuUtilizationList - List of Utilization history Physical Machine
 Y - Predicted CPU utilization of host
 $P_{\text{threshold}}$ - Constant that can be determined experimentally

Gradient Calculation

$$\widehat{\theta}_1 = \frac{1}{m} \sum_{i=1}^m ((\theta_1 + \theta_2 X_i) - Y_i)$$

$$\widehat{\theta}_2 = \frac{1}{m} \sum_{i=1}^m (((\theta_1 + \theta_2 X_i) - Y_i) X_i)$$

X_i, Y_i are selected randomly

The host is overloaded with the computing resources if the CPU utilization is more than 90%, then the corresponding host is represented as an overloaded host. The Linear Adam overloaded detection algorithm is more efficient in predicting the overloaded host than the traditional approaches (Median Absolute Deviation (MAD), Linear Regression (LR), and Ordinary Least Square (OLS)). Initially, the CPU utilization of a single host is evaluated which is provided as input for analysis. Some default parameters ($\alpha, \beta_1, \beta_2, \epsilon, \text{MaxIteration}$) are assigned with a default value and then coefficient parameters (θ, X, Y) are initialized which is easily configurable. The Host threshold is assigned with the value of 0.85 [18]. $\widehat{\theta}_1, \widehat{\theta}_2$ gradient values are calculated. Linear Adam updates the exponential moving range of gradient \widehat{m}_t and squared gradient \widehat{v}_t with the hyper-parameters $\beta_1, \beta_2 \in [0, 1]$. The moving range determines the first moment and raw moment which is controlled by the algorithm thereby preventing the exponential decay rates of the moving average. After collecting the *CpuUtilHistory* from the host and its length is calculated. If the *CpuUtilHistory* length is lesser than 12, then it will return the *host.getTotalRequestMips()* which is greater than product $0.90 * \text{host.getTotalMips}()$ else reversing process is carried on for collecting the *CpuUtilHistory*[]. After gathering the *CpuUtilHistory* iteration takes where X_i contain the CPU current data and Y_i contains the future data. Till *MaxIteration* completion, $g_t, m_t, v_t, \widehat{m}_t, \widehat{v}_t, \theta_t$ values are predicted. ρ assign as 0.5 which is an Adam safety parameter. At last, Y is predicted whose value is greater than $P_{\text{threshold}}$ then the result returns true which means the host is overloaded or else the host is not yet overloaded.

3.1.2. Host Underloaded Detection

To detect the Host under loading condition, a Interquartile Range (IQR) method is implemented which find the difference between the first QU1 and third QU3 quartiles in equation (1).

$$IQR = QU3 - QU1 \quad (1)$$

Lower threshold utilization T_l in equation (2),

$$T_l = 0.4(1 - sp.IQR) \quad (2)$$

$sp = \text{Safety parameter of VM consolidation}$

The lower threshold of CPU utilization is calculated according to equation (2) by multiplying the IQR and sp values. By considering the T_l value, the underloaded host is detected. The pseudo-code of the Host under loading detection is explained below,

ALGORITHM: IQR_HOST_UNDERLOADED_DETECTION

Input: Host,

Output: Boolean,

1. $T_l = 0.4(1 - sp.IQR)$
 2. **return** $host.getUtilization < T_l$
-

Initially, the host is an analysis based on resource utilization and evaluates the lower threshold T_l value. T_l is a product of IQR value and safety parameters of VM consolidation. If the host resource utilization is less than T_l then it is assigned as an underloaded host.

3.1.3. VM Selection (MMT)

In VM selection, we utilized Minimum Migration Time (MMT) policy [4] for selecting the VMs. Based on the minimum migration time of VM which is transacted among the source host to the target host. To prevent the host Overloaded, the MMT policy is iterated simultaneously.

3.1.4. VM Allocation (PABFD)

In VM Allocation, Power-Aware Best Fit Decreasing (PABFD) algorithm [14] for VM placement sort the VM based on CPU utilization in descending order. It is a basic algorithm in the CloudSim simulator which sorted out the low power utilized VMs. It chooses the least power utilizing VM which is compared with the aggregate power of Hosts. The total host and VM list are allocated to PABFD, the minimum power utilized VM is selected and analysis the host resources for VM. Then the power difference among the host and VM is evaluated then the corresponding host is allocated with the selected VM. After the VM allocation, those VMs are migrated to the destination host.

4. Results and Discussion

This section describes the Simulation environment and evaluated parameters to represent the performance of the proposed approach compared with the traditional approaches in terms of energy consumption, SLA degradation, Number of VM migration, and Number of host shutdowns.

4.1. Simulation Environment

The CloudSim-3.0.3 toolkit is used as the simulation environment for implementing the load balancing prediction in the cloud data center. We have taken 800 Physical Machines (PMs) which are categorized into two sections for evaluating the VM migration. 1)The PM category1 uses HP ProLiant ML110 G4 model, Intel Xeon 3040 processor, 4GB memory for each PM, and 2CoresX1860 MHZ processor cores per PM. The PM category2 uses HP ProLiant ML110 G5 model, Intel Xeon 3075 processor, 4GB memory for each PM, and 2CoresX2660 MHZ processor cores per PM. Network Attached Storage (NAS) is utilized to store the VM files and user's data where nearly 500 data centers are available. We have implemented the novel Load Balancing Approach in real workload traces of data center servers. It is applicable in a real cloud environment which makes the proposed approach a versatile one. PlanetLab and random workloads dataset are used for testing the proposed load balancing approach which predicts its efficiency. At last, it is tested under the PlanetLab which has a largest trace of VMs and Random datasets.

4.2. Simulation Results

To investigate the performance of the proposed load balancing approach, we evaluated some metric parameters such as energy consumption of datacenter, SLA degradation, Number of VM migration, and Number of host shutdowns which are depicted in Fig 2. When the cloud provider is unable to satisfy the resource requirement of the users then it will cause an SLA violation. It is calculated based on two different independent metrics such as SLA violation time per active host (*SLATAH*) and performance degradation during migration (*PDM*) (equation 3). The *SLATAH* and *PDM* is calculated based on equations (4), (5),

	Workload	Number of Migration	Energy (Kwh)	SLA Degradation (%)	Host shutdown
20110303	Linear Regression	9193	101.1517	0.00029280	1035
	Linear Adam	1489	99.8749	0.00003925	785
	MAD	8600	127.9658	0.00026240	1041
20110306	Linear Regression	6932	75.916	0.000297479	1020
	Linear Adam	1297	67.8144	0.00004064	794
	MAD	6282	96.9644	0.00026777	1000
20110309	Linear Regression	9110	85.98381	0.00042006	1051
	Linear Adam	3073	68.9172	0.00012356	830
	MAD	22603	89.43056	0.00012989	1143
Random	Linear Regression	991	29.5238	0.000539598	216
	Linear Adam	85	20.4822	0.00006184	55
	MAD	2732	34.5607	0.00215343	1041

Table.1. Simulation result of Linear Adam, LR and MAD algorithms with different workload

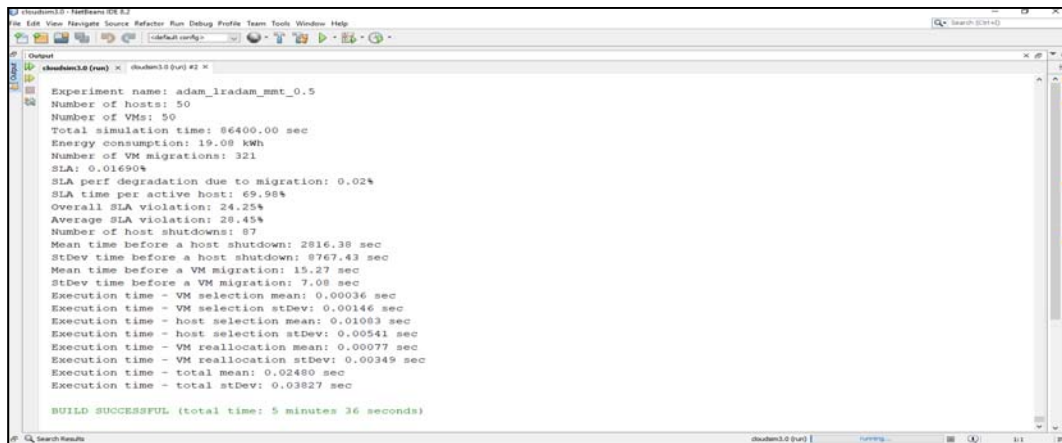


Fig 2: Simulation result of 20110303 workload

$$SLA\ violation = SLATAH \times PDM \quad (3)$$

$$SLATAH = \frac{1}{No.of\ Host} \sum_{k=1}^{No.of\ Host} \frac{T_U}{T_a} \quad (4)$$

$$PDM = \frac{1}{No.of\ VMs} \sum_{k=1}^{No.of\ VMs} \frac{PDM}{T_{cu}} \quad (5)$$

T_U = total period of time the user experiences a 100% resource utilization

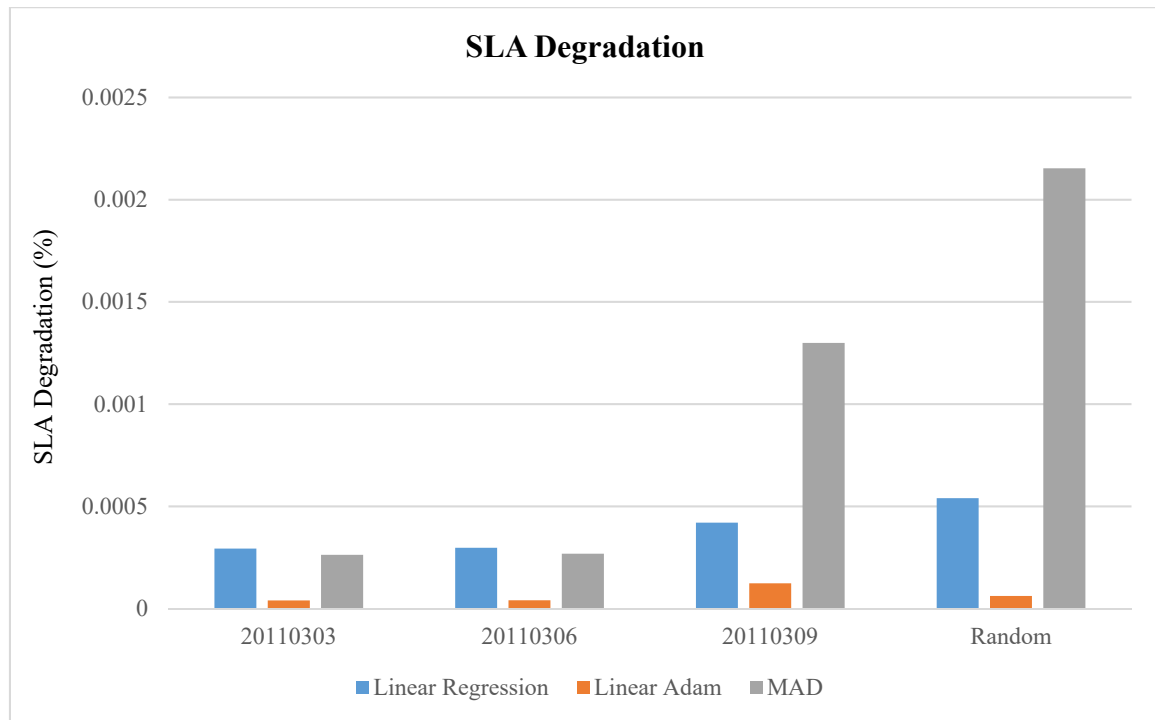
T_a = total period of time the host is in active state

N = total number of hosts

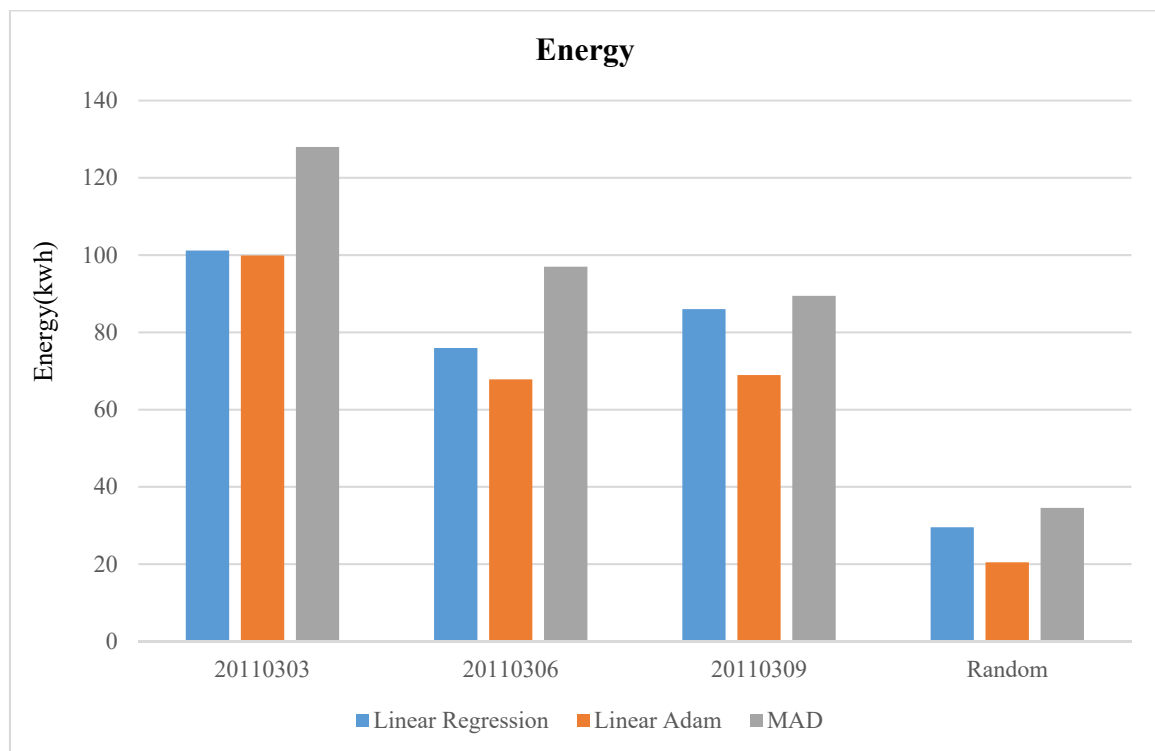
v =total number of VMs

PD_m =performance degradation of VM during migration

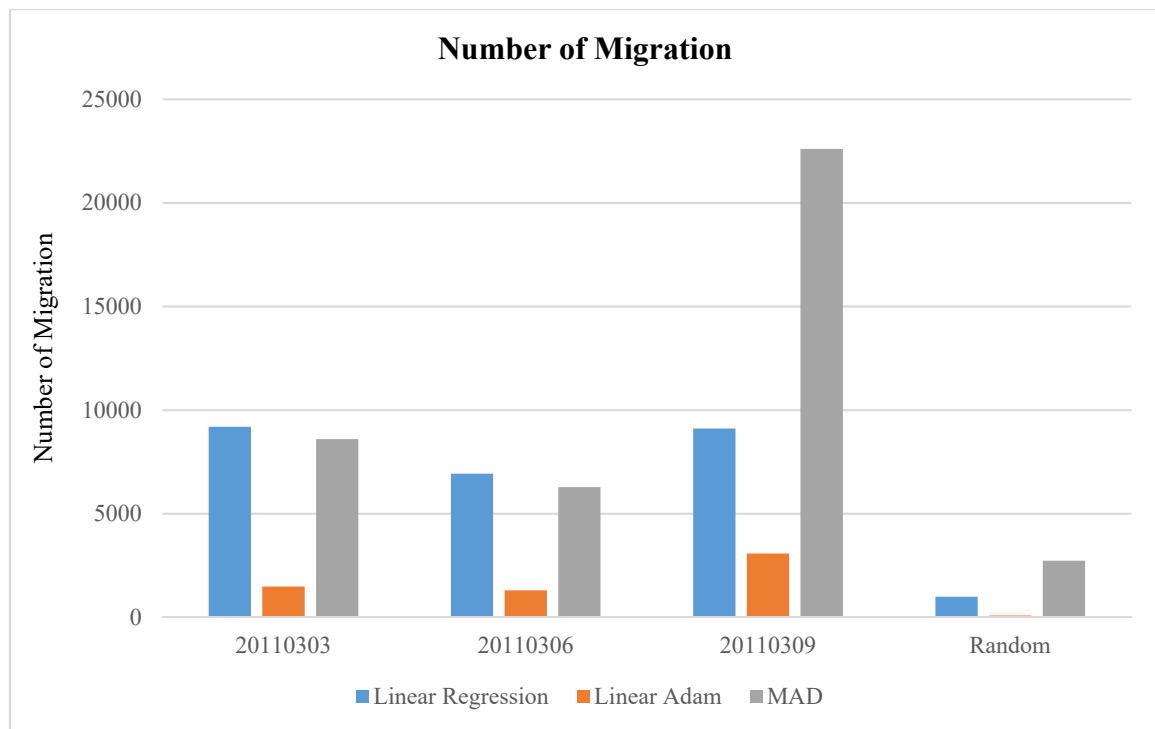
PD_m =total CPU capacity requested by VM



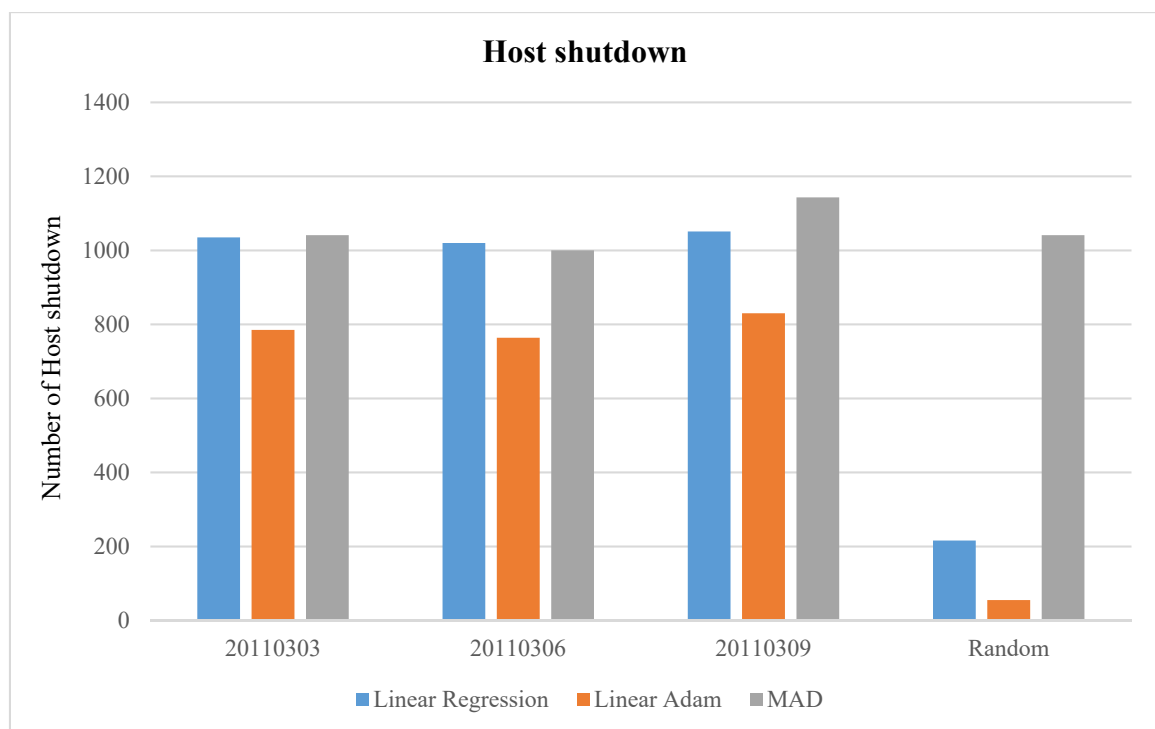
a)



b)



c)



d)

Fig 3: Performance analysis of proposed load balancing approach, LR, and MAD in terms of a) SLA Degradation b) Energy Consumption c) Number of Migration d) Host Shutdown

We calculated the Average SLAV, overall SLA violation, Number of VM migration, and Number of host shutdowns represented in equation (6)(7)(8)(9).

$$Average\ SLAV = \frac{\sum_{k=1}^v (requestedMIPS) - \sum_{k=1}^v (allocatedMIPS)}{v} \quad (6)$$

$$\text{Overall SLAV} = \frac{\sum_{k=1}^v (\text{requestedMIPS}) - \sum_{k=1}^v (\text{allocatedMIPS})}{\sum_{k=1}^v (\text{requestedMIPS})} \quad (7)$$

$$\text{Number of Migrations } (H, t_1, t_2) = \sum_{i=1}^N \int_{t_1}^{t_2} \text{Migi}(H) \quad (8)$$

$$\text{Number of Host Shutdown}(HS) = \frac{1}{n} \sum_{i=1}^n hs_i \quad (9)$$

$\text{Migi}(H)$ =no. of host migration from t_1 to t_2

hs_i =number of active hosts at the time i

HS =no.of host shutdown

n = n times (iteration)

Using those metric parameters, we have simulated results of the proposed load balancing approach, LR [15] and MAD [16] which are compared on different workloads and tabulated in Table 1. In performance analysis, we used three different workloads (20110303, 20110306, and 20110309) which are simulated in CloudSim using java code. For each workload, the simulation time, energy consumption, SLA degradation is varied and minimized in the proposed load balancing approach than existing methods (LR and MAD). In the random dataset, consider we took the 20110303 workloads which were allocated with 50 host and 50VMs. The total simulation time is 86400.00 sec, the energy consumption is 19.08 kWh, total VM migration is 321, and SLA degradation value is 0.01690% which is depicted in Fig 2.

The graphical representation of the overall performance of the three methods (Linear Adam, LR, and MAD) is depicted in Fig 3. In SLA Degradation, Linear Adam (0.00003925) provides a lower SLA degradation than LR (0.00029280) and MAD (0.00002624). The energy consumption of Linear Adam (99.8749) is lesser than the LR (101.1517) and MAD (127.9658). Similarly, the VM migration in linear Adam (1489) is also lower than the LR (9193) and MAD (8600) algorithms. At last, we evaluated the number of host shutdown needed for proceeding the VM migration, it is shown that Linear Adam (785) require less number of host shutdown than LR (1035) and MAD (1041). Based on the computational metrics, it can be concluded as a result that the proposed novel load balancing approach can provide an optimal solution for improving the QoS between cloud providers and users.

5. Conclusion

In this paper, we proposed a novel approach towards cloud computing by balancing the load among the host in an efficient way. Based on research analysis, the cloud computing field requires less energy consumption and versatile resource provision to the customer. Load balancing is a main key area that ensures scalability, reliability and minimizes response time in cloud datacenter that improvise the computation and QoS. This paper deal with the broad perspectives of load balancing approaches and provides a good trade-off performance through Linear Adam, IQR, and MMT policy. Linear Adam can solve all high dimensional complex problem which is used to predict the overloaded host, IQR method is highly robust in underloaded host detection, MMT for evaluating the VM which acquire the minimum time of migration from source to the destination host, and PABFD sort all the VM based on CPU utilization. We conducted a simulation experiment on the proposed load balancing, LR, and MAD approaches using Real PlanetLab and Random workloads for evaluating the performance metrics. Hence, our proposed approach will be the promising solution for diminishing the energy consumption of PM and SLA degradation among users. As future work, we plan to reduce the SLA Violation Time per Active Host (%) by implementing an optimal resource allocation method that reduced the SLA violation and overcomes the VM's resource demands conflict.

References

- [1] Johns and Bartlett, Boston. Jones Luis Schaefer & Julio Cezar Mairesse Siluk & Patricia Stefan de Carvalho & José Renes Pinheiro & Paulo Smith Schneider. (2020). *Management Challenges and Opportunities for Energy Cloud Development and Diffusion*, Energies, MDPI, Open Access Journal, vol. 13(16), pages 1-27, August.
- [2] Lakra, Atul & Yadav, Dharmendra. (2015). *Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization*. Procedia Computer Science. 48. 107-113. 10.1016/j.procs.2015.04.158.
- [3] Mapetu, Jean-Pepe & Kong, Lingfu & Chen, Zhen. (2021). *A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing*. The Journal of Supercomputing. 77. 1-42. 10.1007/s11227-020-03494-6.
- [4] Mandal, R., Mondal, M.K., Banerjee, S. (2020). *An approach toward design and development of an energy-aware VM selection policy with improved SLA degradation in the domain of green cloud computing*. J Supercomput 76, 7374–7393. <https://doi.org/10.1007/s11227-020-03165-6>
- [5] Ibrahim M, Imran M, Jamil F, Lee Y-J, Kim D-H..(2021). *EAMA: Efficient Adaptive Migration Algorithm for Cloud Data Centers (CDCs)*. Symmetry. 2021; 13(4):690. <https://doi.org/10.3390/sym13040690>
- [6] Haghshenas, K., Mohammadi, S. (2020). *Prediction-based underutilized and destination host selection approaches for energy-efficient dynamic VM consolidation in data centers*. J Supercomputer 76, 10240–10257 <https://doi.org/10.1007/s11227-020-03248-4>

- [7] Alhammadi, Akram Saeed Aqlan & Varadharajan, Vasanthi. (2020). *MR-MOSLO: VM Consolidation Using Multiple Regression Multi-Objective Seven-Spot Ladybird Optimization for Host Overload Detection*. International Journal of Intelligent Engineering and Systems. **13**. 20-30. 10.22266/ijies2020.0430.03.
- [8] Yadav, R., Zhang, W., Li, K. (2021). *Managing overloaded hosts for energy-efficiency in cloud data centers*. Cluster Comput (2021). <https://doi.org/10.1007/s10586-020-03182-3>
- [9] A. El-Moursy, A., Abdelsamea, A., Kamran, R. (2019). *Multi-Dimensional Regression Host Utilization algorithm (MDRHU) for Host Overload Detection in Cloud Computing*. J Cloud Comp **8**, 8. <https://doi.org/10.1186/s13677-019-0130-2>
- [10] Masdari, M., Khezri(2020), H. *Efficient VM migrations using forecasting techniques in cloud computing: a comprehensive review*. Cluster Comput **23**, 2629–2658. <https://doi.org/10.1007/s10586-019-03032-x>
- [11] Priyanka Nehra, A. Nagaraju. (2021). *Host utilization prediction using hybrid kernel-based support vector regression in cloud data centers*. Journal of King Saud University – Computer and Information Sciences. <https://doi.org/10.1016/j.jksuci.2021.04.011>
- [12] A. K. Kulkarni and B. Annappa.(2019). *Context Aware VM Placement Optimization Technique for Heterogeneous IaaS Cloud*, in IEEE Access, vol. **7**, pp. 89702-89713, , doi: 10.1109/ACCESS.2019.2926291.
- [13] <https://towardsdatascience.com/understanding-gradient-descent-and-adam-optimization-472ae8a78c10>
- [14] Beloglazov, Anton & Buyya, Rajkumar. (2012). *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**. 10.1002/cpe.1867.
- [15] N.Nirmala Devi,S.Vengatesh Kumar. (2019). *A Study on Virtualization in Cloud Computing*, published in International Journal of Communication and Networking System, Volume: 08 Issue: 02 December 2019 Page No.70-75, ISSN: 2278-2427
- [16] Mandal G., Dam S., Dasgupta K., Dutta P. (2020) *A Linear Regression-Based Resource Utilization Prediction Policy for Live Migration in Cloud Computing*. In: Mandal J., Mukhopadhyay S., Dutta P., Dasgupta K. (eds) Algorithms in Machine Learning Paradigms. Studies in Computational Intelligence, vol **870**. Springer, Singapore. https://doi.org/10.1007/978-981-15-1041-0_7
- [17] Jararweh, Yaser & Bani Issa, Manar & Daraghme, Mustafa & Al-Ayyoub, Mahmoud & Alsmirat, Mohammad. (2018). *Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation*. Sustainable Computing: Informatics and Systems. **19**. 10.1016/j.suscom.2018.07.005.
- [18] M. Daraghme, S. Bani Melhem, A. Agarwal, N. Goel and M. Zaman (2018). *Linear and Logistic Regression Based Monitoring for Resource Management in Cloud Networks*, 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), 2018, pp. 259-266, doi: 10.1109/FiCloud.2018.00045.

Authors Profile



Nirmala Devi N has completed Master of Philosophy at June 2013 from D.K.M. College for Women (Autonomous), Thiruvalluvar University, Vellore, India. Her dissertation work was carried out in Cloud Computing environment. She has completed Master of Computer Application at June 2007 from Institute of Road and Transport Technology, Anna University, Erode, India. She was studied B.Sc. Computer Science at April 2004 in Marappan Lakshmiammal Arts & Science College, Madras University, Tirupattur, India. She is pursuing Ph.D. Computer Science in Dr.SNS RajaLakshmi Arts and Science College (Autonomous), Bharathiar University, Coimbatore, India from July 2013 to till date. She is currently working as a Assistant Professor, Department of Computer Science in Auxilium College (Autonomous), Vellore, India. Her dissertation work is carried out in Virtual Machine Migration in the field of Cloud Computing under the guidance of Associate Prof. Dr. Vengatesh Kumar of Dr.SNS Rajalakshmi College of Arts and Science, Coimbatore, India.



Vengatesh Kumar S has completed B.Sc. Computer Science and M.C.A. in Bharathiar University, Coimbatore, India in 1994 and 1997 respectively. He has received M.Phil. Computer Science and Ph.D. Computer Science from Manomaniam Sundarnar University, Thirunelveli, India in 2003 and 2017 respectively. At present he is working as a Associate Professor, Dept. of Computer Applications in Dr.SNS Rajalakshmi College of Arts & Science, Coimbatore. He is currently reviewer for Springer journals and member in International Association of Engineers (IAENG) & ACM. His research interest includes Cloud Computing, Image Processing and Data Mining applications.