www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC
Gestão, Inovação e Tecnologias

Threshold Compare and Load Balancing Algorithm to for Resource Optimization in a Green Cloud

Nagamani H Shahapure[1]; P.M. Rekha[2]; N. Poornima[3]

[1]Assistant Professor, Department of Information Science, JSS Academy of Technical Education, Bangalore, India.

[1]nagamanishahapure@gmail.com

[2]Associate Professor, Department of Information Science, JSS Academy of Technical Education, Bangalore, India.

[3]Associate Professor, Department of Electronics and Communication, JSS Academy of Technical Education, Bangalore, India.

Abstract

The reduction in the consumption of energy by the cloud data centers is called as green cloud. Green cloud also helps in restraining the waste disposed to the environment. With the increasing demand in cloud computing, there has been an increase in the energy consumption. Green cloud can be achieved by server consolidation and proper load balancing techniques using virtual machine (VM) migration. It is a feature provided by virtualization. The virtual machines are transferred from one host to another. The overhead associated with migration is performance degradation. This can be overcome by proper load balancing techniques. This helps to curtail the number of VM migrations as well as the energy consumption. In this paper, a technique called as threshold compare and load balance algorithm (TCLBA) is proposed for optimization of the resources at the cloud provider. Two types of threshold are defined for load balancing, namely the lower and upper threshold. This algorithm works on the principle of shifting the load from a server if the load is above the upper threshold or shifting to the server if its load is below the lower threshold. The load is balanced by migrating the VMs. The workload is consolidated to smaller number of hosts such that the remaining hosts are shut down. This method solves the purpose of effective utilization of available resources with lesser energy consumption.

Key-words: Green Cloud, Load Balancing, VM Migration, Resource Management, Server Consolidation.

1. Introduction

Cloud offers economical and reasonable solutions to different types of computing by allowing the clients to access scalable resources at any time [1]. Cloud data centers manage inbound tasks by

Received: 07.07.2021 – Accepted: 05.08.2021

provisioning resources [2]. The fundamental technologies to manage data centers are resource allocation and resource scheduling. The carbon emission is reduced, resource utilization is improved and load is balanced in this method. The distribution of the resources is taken care by the cloud system [3]. Therefore, it becomes essential to find solutions for utilizing energy efficiently and reduce the impact of the carbon emission. Many solutions were proposed in this regard, and most of them were based on the concept of server consolidation.

Virtualization improves resource utilization by creating multiple instances of virtual machine (VMs) on a physical server [4]. Virtual machine monitor (VMM) takes care of all the activities of the VM. VM migration is responsible for the resources required by the VM which is dynamic in nature. VM migration comprises of shifting the VM from one host to another. This deals with load balancing, server consolidation, server failures. Performance degradation is the drawback of migrations. To provide good services, the number of VM migrations need to be reduced by designing an experimental algorithm for server consolidation [5]. The increasing demands of the computational power is met by constructing large scale virtualized data centers. This increases the energy consumption which effects the environment [6].

In this paper, the technique for server consolidation using threshold compare and load balance algorithm (TCLBA) is discussed. It works based on the threshold values. The objective of the proposed algorithm is optimization of the cloud data center by balancing the load on the physical machines. This method consumes less energy with minimum SLA violations. Organization of the paper is as follows. Section 2 discusses about the related work. Section 3 discusses the background knowledge of the server optimization with load balancing and server consolidation. Section 4 and 5 elaborates the basic design and methodology of the proposed algorithm respectively. Section 6 discusses about the experimental simulation and results. The conclusion and future work is summed up in section 7.

2. Literature Review

Lei Xie [7] proposed a novel self-adaptive VM consolidation strategy. The PMs (Physical Machine) are selected based on dynamic multi thresholds. This approach gives good QoS and also utilizes less energy. In addition, the benefit of this method is the reduction on the number of active hosts especially under extreme workloads. The other parameters like energy efficiency and SLA violations for migration has not been considered.

Tsai et al. [8] suggested an enhanced differential evolution algorithm. This method is based on cost and time model. The drawback of this method is variations in the tasks are not taken into consideration.

Gawali, M.B. and Shinde, S.K [9] proposed a heuristic procedure for allocating the resources in cloud environment in an effective manner. The computing resources are utilized efficiently. In this method response time and processing time are not considered.

Cheng et al. [10] presented task scheduling algorithms which is based on a vacation queuing model. The problem of proper utilization of the resources like network, servers, storage is not analyzed in this approach.

Gupta, S., Tiwari, D. and Singh, S [11] presented a VM migration based load balancing method. Lower and upper threshold values of the workload are defined to balance the load. If the workload exceeds the upper threshold or is lesser than the lower threshold, system is said to be imbalanced and the VMs are migrated.

Lin et al. [12] proposed method of scheduling parallel workloads. This method used the first come first serve (FCFS) approach which orders the jobs based on the availability of the resources. This system does not emphasis on terminating the tasks and starvation of the PMs.

Yadav, R. and Zhang, W [13] proposed an adaptive energy-aware algorithm. VM selections are defined to consolidate the VMs from overloaded or underloaded host. The objective of this approach is minimization of energy utilization without affecting the QoS. It also minimizes the service level agreement (SLA) violations.

Kaur, B., Kaur, N. and Singh, R [14] states green data centers can be implemented by minimizing energy consumption in cloud computing.

Rodriguez MA and Buyya R [15] proposed a method for reduction of execution costs. This method uses meta-heuristic and particle swarm optimization.

Darwish, R.R. and Elewi [16] have presented a framework that is a combination of a heuristic approach with a control theory method for a green cloud. The proposed technique resulted in reducing the total energy consumption maintained the negotiated service level of web applications.

Zhou *et al.* [17] proposed an Adaptive Three-threshold Energy-aware (ATEA) algorithm to decrease energy and SLA violation in cloud data centers. Based on underloaded, moderately loaded, overloaded and little loaded hosts, resource scheduling takes place.

3. Background

Virtual machine migration helps in balancing the load on a host machine. Figure 1 and figure 2 depicts the unbalanced and balanced cloud data center respectively. Figure 3 and figure 4 respectively represent the situation without server consolidation and with server consolidation. Figure 1 shows a data center with different work load. The work load is depicted in terms of virtual machines. There are some PMs which are heavily loaded and some are underloaded. To increase the performance of the data center, the physical machines should be effectively managed such that the response time of the services is also improved. This implies, if the load in the host is balanced then the response from the server to the client is faster. Server consolidation is an important approach to manage the workload by proper management of the resources. It reduces the total number of servers required by an organization to save energy. Figure 3 depicts a situation where cloud data center is underloaded. The server consolidation method optimizes the cloud server by transferring the load to a single host that is lightly loaded. This reduces the number of active machines. It is done using VM migration technique. After the migration the hosts are switched off to save energy.

Fig. 1 - Unbalanced Cloud Data Center

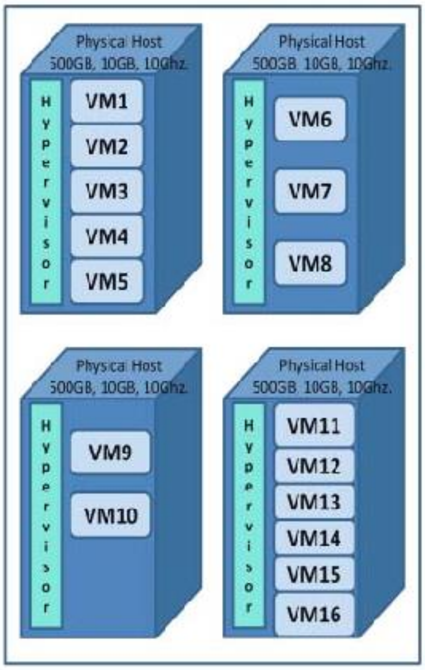


Fig. 2 - Balanced Cloud Data Center

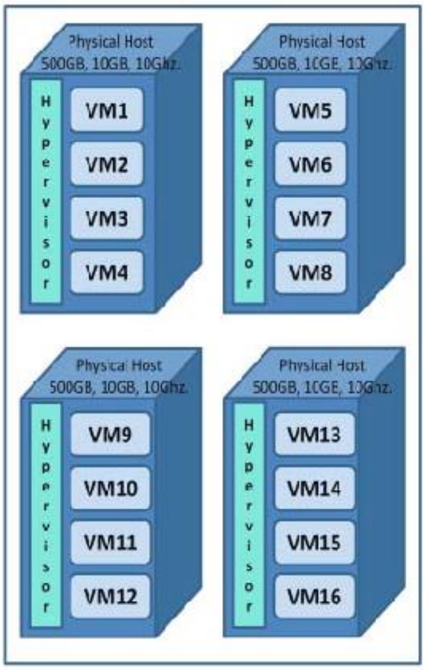
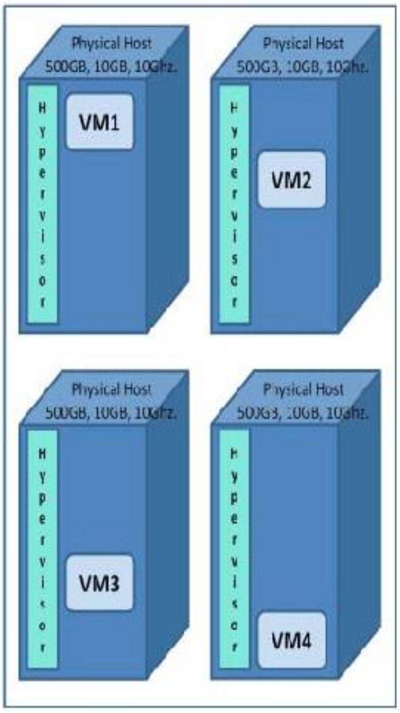
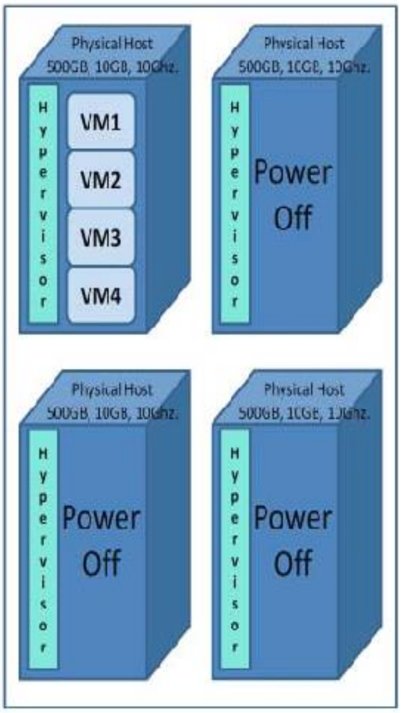


Fig. 3 - Data Center without Server Consolidation; Fig. 4 - Data Center after Server Consolidation



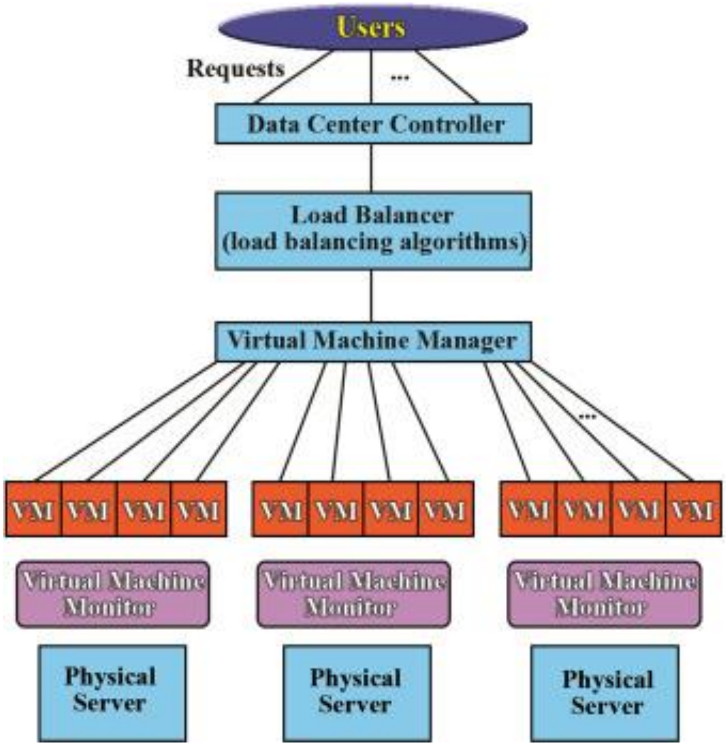


Through consolidation of servers, the number of active physical servers as well as power usage of cloud operations are reduced. This implements the concept of green computing. Server consolidation also maintains the QoS without any compromise in the SLA.

a. Cloud Optimization

The cloud system is represented in figure 5. The data center controller manages the workload and responds to the client requests. The load balancer takes care of distributing the traffic across multiple servers. The virtual machine manager runs several operating systems concurrently on a single physical host. The virtual machine monitor (VMM) is a software program that responsible for creation and management of a virtual machine. The requests are first arrived at the controller. The controller along with the load balancer and the VM manager distributes the request to the necessary VM on the PM. This is done such that the load is balanced among all VMs on the PMs.

Fig. 5 - Cloud System Architecture



4. Proposed Work

Every PM in the cloud hosts several VMs. These VMs allow several applications to run on a host machine. The VM load is reliant on the application and size. The PM load is the total load of all VMs executing on it. The migration of the VMs is dependent on a threshold value. The threshold value can be static or dynamic. Lower and upper thresholds are predetermined and they remain fixed in static method. In dynamic threshold, both the thresholds are bound to change during the execution time. The demand of the resources by the VMs vary dynamically; therefore dynamic threshold is appropriate for the cloud. There is no pre knowledge about the workload which are submitted by different users.

The proposed approach is based on dynamic calculation of the threshold. There are two types of thresholds that are used. The first one is the upper threshold which describes the overloaded condition of the hosts. The second one describes the lower threshold that defines the underloaded situation. The proposed method aims to lessen the number of migration and also minimize the energy consumption. Following are the steps involved for server consolidation.

1. Determine the load of PM and VM.
2. Evaluate overloaded and underloaded condition based on the threshold.
3. Migrate a VM that ensures energy efficiency, avoids SLA violations and preserves the QoS parameters.

4. Select the appropriate PM for VM placement.

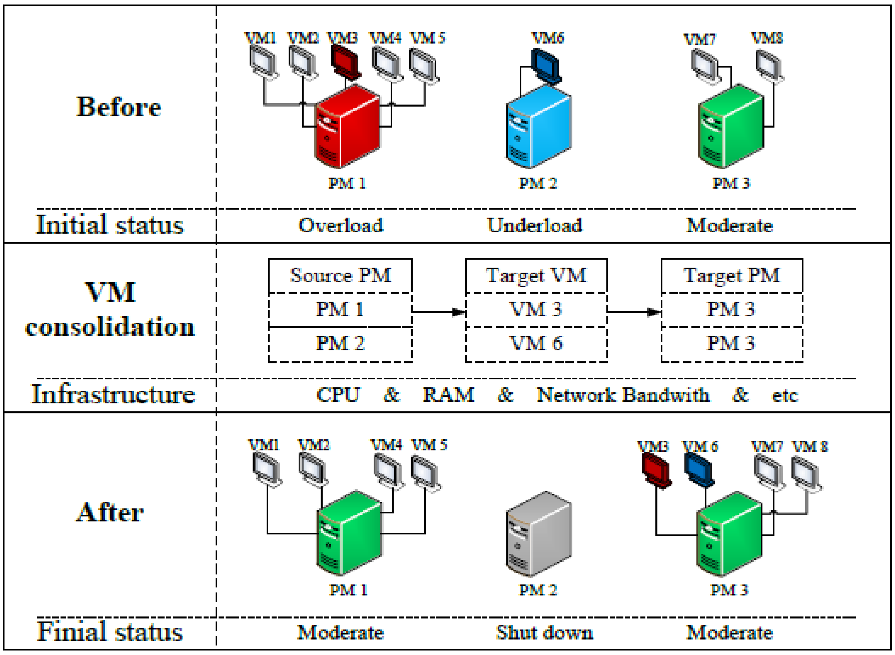
a. VM Consolidation

VM consolidation comprises of the three below steps:

- Source PM Selection: Selection of PM depends from where the VMs are migrated. This PM is considered as the source PM.
- Target VM selection: This module selects the VMs from a source PM. The source PM is the one that was selected in the previous step as input.
- VM placement: Here a PM is selected to hold the VMs that are migrated from the previous step or the new VM requests.

The server consolidation architecture is shown in figure 6. The VMs are spread across multiple PMs before the VMs are consolidated. The PM 2 is underloaded with only one VM, while PM 1 is overloaded with 5 VMs. The overloaded PM can cause SLA violations as it may hinder the QoS attributes like response time and processing time. On the other hand the underloaded PM has to remain alive because of the single VM it is hosting, resulting in energy consumption. To avoid SLA violations and reduce energy consumption, VM consolidation is required.

Fig. 6 - Architecture of the Server Consolidation



The PMs are either overloaded, moderate or underloaded, depending on the threshold values. VM consolidation strategy results in two live servers in the system with adequate load. In addition, the underloaded server (PM 2) is shut down resulting is lesser energy consumption.

b. System Architecture

Cloud data center consisting of n heterogeneous PMs, PM = {PM_1, PM_2… PM_n} is considered. The resources in the PM are CPU, RAM and network bandwidth. The hosts contain m number of virtual machines, VM = {VM_1, VM_2…… VM_m}. The PM also contains VMM. The allocation of VMs to the PMs is done by the VMM.

1. User submits request, demanding for resources like CPU, storage, network on the cloud.

2. The data center controller along with the load balancer optimize the placement of VMs for their consolidation.

3. Migration of VMs along with their power adjustments is taken care by the VMM in each node.

The usage of the processor, RAM, hard disk, and bandwidth are few of the parameters considered for energy consumption of the servers. Energy consumption is dependent on the total power consumption and CPU utilization. Server's power consumption increases as the CPU changes its state from idle to busy. The power consumption by the server is calculated by a linear function of the current CPU utilization (UTIL) as:

$$Pow_{TOT} = (Pow_{BUSY} - Pow_{IDLE}) * UTIL + Pow_{IDLE} \qquad (1)$$

Where Pow_{TOT} is the projected power consumption, Pow_{BUSY} is the power consumption value when a server is completely consumed and Pow_{IDLE} is the power consumption value when a server is idle.

c. Formulations and Assumptions

Assume a data center with comprising of virtual machines. The VMs in turn consists of cloud resources. The steps for server consolidation are:

1. Initiation of VM Migration
2. Source VM Selection
3. Target VM and PM selection.

The problem formulation is given in Table 1. A physical server can be uniquely identified in the form of PM(n) = {VM_{LIST}, PM_{UTIL}, Status}, where VM_{LIST} is the list of all VMs on PM(n). PM_{UTIL} represents the resource utilization on PM(n).

Table 1 - Notations

Notations	**Descriptions**
VM(m)	The mth VM
PM(n)	The nth PM
PM_{LIST}	List of PMs
VM_{LIST}	List of VMs
TH_{SLA}	Threshold level of SLA
VM^{*}_{j}	Utilization ratio of parameter specified by * on jth VM
PM^{*}_{i}	Utilization ratio of parameter specified by *on PM(i)
TH_{LOW}	Existing lower threshold on parameter *
TH_{UP}	Existing upper threshold of the PM
TH_{CPU}	Existing CPU threshold on the PM
TH_{RAM}	Existing RAM threshold on the PM
TH_{BW}	Existing bandwidth threshold on the PM
VM_j^{CPU}	Allocated CPU on VM(j)
VM_j^{NET}	Allocated bandwidth on VM(j)
VM_j^{RAM}	Allocated RAM on VM(j)
PM_i^{CPU}	Allocated CPU on PM(i)
PM_i^{NET}	Allocated bandwidth on PM(i)
PM_i^{RAM}	Allocated RAM on PM(i)
$Source_{PM}$	Source PMs from where VMs are migrated
$Select_{VM}$	VMs selected for migration
$Target_{PM}$	List of destination PMs to which the VMs are shifted

For reducing the energy consumption the calculation of the threshold is based on the equation below.

$$T = \{TH_{LOW}, TH_{UP}, TH_{SLA}\} \quad (2)$$

Keep a check on the threshold of the SLA. The list of the threshold values of all the PMs is maintained. If the current SLA > threshold SLA then the SLA is being violated. Too much violation in SLA will lead to low QoS. To handle such situations reduce the workload on the overloaded PMs. This can be done by migrating the workload to the PMs which are less loaded. This can be done by checking the PMs whose current SLA < threshold SLA.

5. Methodology

The process of shifting the VM from source to target is called migration.

a. Initiation of VM Migration

A threshold value is used to identify whether the source PM is overloaded or underloaded. Migration is carried out in both the situations.

Calculation of Load on the PM and VM

The load calculation depends on three parameters i.e. CPU, memory and network bandwidth. Each VM is assumed to have its own CPU, memory and bandwidth. VM load is calculated as:

$$VM.CPU_{UTIL} = \frac{\sum VM_j^{CPU}}{\sum PM_i^{CPU}} \tag{3}$$

$$VM.NET_{UTIL} = \frac{\sum VM_j^{NET}}{\sum PM_i^{NET}} \tag{4}$$

$$VM.RAM_{UTIL} = \frac{\sum VM_j^{RAM}}{\sum PM_i^{RAM}} \tag{5}$$

$$\text{VM}_{\text{UTIL}} = \text{VM.CPU}_{\text{UTIL}} + \text{VM.NET}_{\text{UTIL}} + \text{VM.RAM}_{\text{UTIL}} \tag{6}$$

$$PM_{UTIL} = \sum_{j=1}^{j=n} VM_{UTIL} \tag{7}$$

VM load is related to the CPU usage of the VM.

$$VM_{LOAD} = VL = \frac{\sum VM_j^{CPU}}{\sum PM_i^{CPU}} \tag{8}$$

The total capacity of the host machine is calculated as the total load of the VM executing in that host. If there are m VM on nth host, then average load on the nth host is calculated by equation 9.

$$PM_{LOAD} = \frac{\sum_{j=1}^{m} VL_j}{m} \tag{9}$$

Upper and Lower Threshold Calculation

There is no pre knowledge about the workload which are submitted by different users. Therefore it is necessary to calculate the upper and the lower thresholds. Equation 10, 11 and 12 gives the calculation of the upper threshold of the CPU, RAM and network bandwidth respectively.

Upper Threshold Calculation

$$TH_{CPU} = \frac{\sum_{j=1}^{m} VM_j^{CPU}}{PM_i^{CPU}} \quad (10)$$

$$TH_{RAM} = \frac{\sum_{j=1}^{m} VM_j^{RAM}}{PM_i^{RAM}} \quad (11)$$

$$TH_{NET} = \frac{\sum_{j=1}^{m} VM_j^{NET}}{PM_i^{NET}} \quad (12)$$

$$TH_{TMP} = \sum(TH_{CPU}, TH_{RAM}, TH_{NET})/3 \quad (13)$$

$$TH_{\boldsymbol{UP}} = 1 - X * TH_{TMP} \quad (14)$$

$$TH_{\boldsymbol{UP}} = 1 - 0.5 * TH_{TMP} \quad (15)$$

Where ‘X’ determines the percentage of the load on the host. The experimental value of X is 5%.

Lower Threshold Calculation

The CPU utilization is lower than 30%; lower threshold is always 0.3. The lower threshold is calculated as in equation 16.

$$TH_{LOW} = PM_{UTIL} \leq 30\% = 0.3 \quad (16)$$

b. Source VM Selection

Migration time and system down time are the parameters to select VMs for migration. In this method the VMs are arranged in descending order.

Algorithm 1: VM Selection

```
Input: $PM_{LIST}$, $VM_{LIST}$
Output: $Select_{VM}$
for each PM(i) in $PM_{LIST}$ do
    if $((PM^{*}_{i} > TH_{UP})$&& $(PM(i)_{SLA} > T_{SLA}))$ then
        $Source_{PM} \leftarrow PM(i)$
        $VM_{LIST}$.sortDescending($VM_{UTIL}$)
        Foreach VM(j) in $VM_{LIST}$ do
            $x = PM_{UTIL} - TH_{UP}$
            $VM\_MIG = VM_{UTIL} - x$
            $Select_{VM} = VM\_MIG$
        End for
    End if
    if $((PM^{*}_{i} < TH_{LOW})$&& $(PM(i)_{SLA} < TH_{SLA}))$ then
        $Source_{PM} \leftarrow PM(i)$
        Foreach VM(j) in $VM_{LIST}$ do
            $x = TH_{LOW} - PM_{UTIL}$
            $VM\_MIG = x - VM_{UTIL}$
            $Select_{VM} = VM\_MIG$
Return selectedVM
```

c. Target PM Selection

Selection of appropriate PM is one of the essential job in load balancing. An incorrect selection of the physical machine can degrade the performance. This algorithm places the VM on a host which consumes lowest power in the list of hosts present in the data center.

Algorithm 1: PM Selection

```
Input: $PM_{LIST}$, $VM_{LIST}$
Output: PM Selection
for each PM(i) in $PM_{LIST}$ do
if $((PM^*_i < TH_{UP}) \&\& (PM(i)_{SLA} < T_{SLA}))$ then
Source $_{PM}$ $\leftarrow$ PM(i)
$VM_{LIST}$.sortDescending($VM_{LOAD}$)
Foreach VM(j) in $VM_{LIST}$ of the Source $_{PM}$ do
Calculate each $VM_{LOAD}$ using (8)
$VM_{LIST}$.sortAscending($VM_{LOAD}$)
If enough resources exist in a PM for VM then
$Target_{PM}$ = PM(i)
minEngy = MAX
allocatPM = NULL
foreach PM in $Target_{PM}$ do
energy = calEnergy(PM) as in (1)
if (energy < minEngy) then
allocatPM = PM
minEngy = energy
if allocatPM $\neq$ NULL then
allocate VM to allocatPM
If ($VM_{MIGRATE}$ == NULL) then
switchOff(use_VM)
return PM Selection
```

6. Performance Evaluation

Performance evaluation method is given below.

a. Workload Data

CloudSim 3.0 is used to create a simulation environment as it allows to model and simulate real-world cloud infrastructure. The workload data for simulation is taken from the CoMon project, which is a monitoring infrastructure for PlanetLab in CloudSim [18]. Table 2 shows the power consumption at different levels. Table 3 and table 4 list the parameters of the PMs and VMs respectively.

Table 2 - Power Consumption at different Load Levels in Watts

Host	Idle	10%	20%	30%	40%	50%	60%	70%	80%	90%	Full
Dell R720	85	87.6	88	88.9	92.5	98	99.8	108	112	114	117
Dell R730	92.8	97	101	102	105	112	118	121	126	132	136

Table 3 - Host Parameters

Parameters	Hosts	
	Dell R720	Dell R730
Number of Hosts	400	400
No of Cores	2	2
MIPS	1870	2670
RAM	4096	4096
BW	1 GB	1 GB
Storage	1.5 GB	2 GB

Table 4 - VM Parameters

Parameters	VM Type			
	High CPU Medium Instance	Extra Large Instance	Small Instance	Micro Instance
No of Cores	1	1	1	1
MIPS	2500	2000	1000	500
RAM	870	1740	1740	613
BW	1 MB	1MB	1 MB	1 MB
Storage	3.85 GB	2 GB	1.75 GB	613 MB

The analysis is done for 20 hour period on 2 different days and the number of VMs that are surveyed on each day is given in table 5.

Table 5 - VMs Used for Simulation

Date	Number of VMs
5 August	1269
8 August	1050

Received: 07.07.2021 – Accepted: 05.08.2021

b. Experimental Setup

A wide range of data center consisting of 800 heterogeneous physical hosts, half of which are Dell R720 2U rack servers, and the other half comprises of Dell R720 XD servers are used for the simulation. The result is compared with Energy Conscious Green Cloud Dynamic Algorithm (ECGCD) [5]. This algorithm reduces the energy consumption and improves the resource utilization. The experimental results is depicted in table 6.

Table 6 - Simulation Results

Workload	Algorithms	Energy (kWh)	Number of migrations
5 August	ECGCD	68.03	10,428
	TCLBA	63.25	9,148
8 August	ECGCD	56.4	8,644
	TCLBA	48.2	7,128
	TCLBA	66.22	10,112

Table 7 gives the comparison of energy consumptions and migrations. The energy consumptions for different workload is shown in figure 7 and 9. The number of VM migrations for various workload traces is shown in figure 8 and 10.

Table 7 - Comparison of Energy Consumption and Migrations

Date		Time in hours	1	3	5	7	9	11	13	15	17
5/8/19	VM Migrations	ECGCD	0	450	8500	1800	3200	4900	6000	7500	9150
		TCLBA	0	434	625	1600	2900	4540	5718	6920	8200
8/8/19		ECGCD	0	350	877	1750	3300	4800	6100	7100	8644
		TCLBA	0	322	612	1300	2500	4130	5500	6500	7128
5/8/19	Energy Consumption	ECGCD	0	5	14.5	28.1	39.8	52.4	57.9	62.1	68.03
		TCLBA	0	4.5	13.4	25.12	35.8	46.54	53.8	57.1	63.25
8/8/19		ECGCD	0	6	20.01	32.1	41.32	47	51.2	53.2	56.4
		TCLBA	0	4.5	16.5	26.21	38.9	40.22	42.1	44	48.2

Figure 7 - Energy Consumption for different Workload

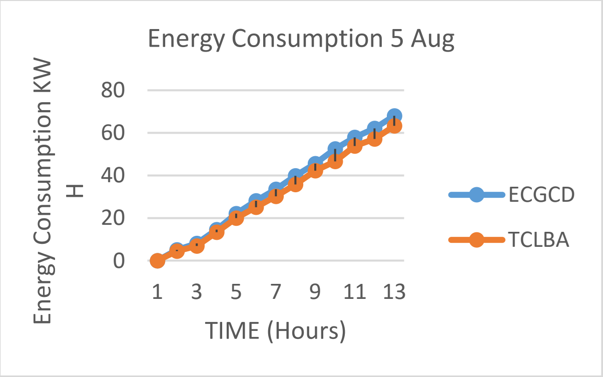


Figure 8 - VM Migrations for different Workload

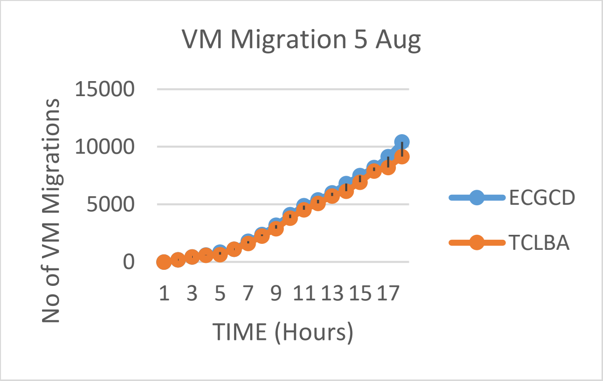


Figure 9 - Energy Consumption for different Workload

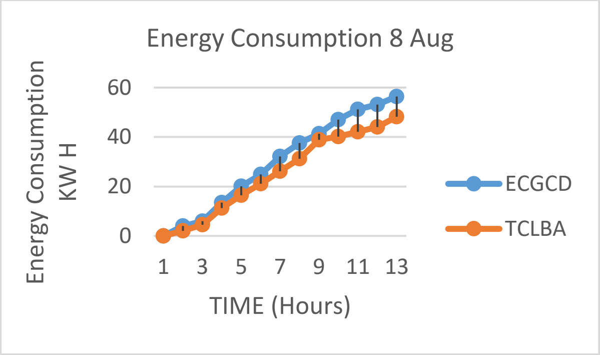
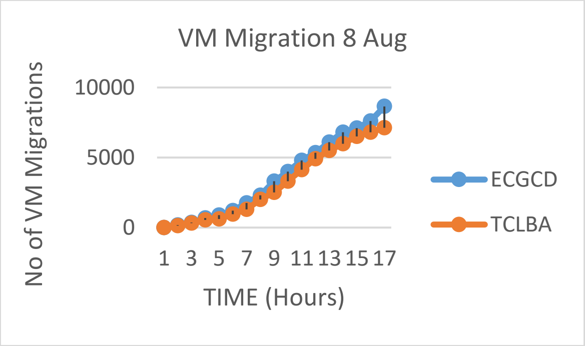


Figure 10 - VM Migrations for different Workload



7. Conclusion and Future Work

This paper shows that proper load balancing and dynamic consolidation techniques is an efficient way to manage the server workload and improves energy efficacy. This technique minimizes the energy consumption by turning off the idle or less utilized servers which enables green cloud. In this paper, both the resource utilization of the hosts and the number of active VMs were considered. Maximum and minimum threshold for the load on the host is calculated. Based on this, the algorithm TCLBA was built. The proposed algorithm was implemented using CloudSim 3.0. The feasibility and effectiveness of the algorithm was tested by running a series of simulations using data extracted from CoMon project. To check the efficiency of the algorithm, it is compared with ECGCD algorithm. The proposed approach reduces the energy consumption as well the number of VM migrations.

Future work will emphasis on more efficient VM migration techniques which will consider the distance between the hosts while implementing the server consolidation and also the SLA violations. This should result in better efficiency of the cloud data center.

References

Jenia Afrin Jeba, Shanta Roy, et al., 2019. Towards Green Cloud Computing an Algorithmic Approach for Energy Minimization in Cloud Data Centers. *International Journal of Cloud Applications and Computing*, 9(1), DOI: 10.4018/IJCAC.2019010105

Mustafa, S., Bilal, K., Malik, S.U.R. and Madani, S.A., 2018. SLA-aware energy efficient resource management for cloud environments. *IEEE Access*, 6, pp.15004-15020.

Xu, X., Fu, S., Cai, Q., Tian, W., Liu, W., Dou, W., Sun, X. and Liu, A.X., 2018. Dynamic resource allocation for load balancing in fog environment. *Wireless Communications and Mobile Computing*, 2018.

Beloglazov, A. and Buyya, R., 2010. November. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. *In MGC@ Middleware* (p. 4).

A.V. Sajitha and A.C. Subhajini, 2017. ECGCD: A Computational Approach of the Designing and Evaluation of Energy Efficiency and Environmental Sustainability in Green Data Centers Using Dynamic VM Consolidation. *Jour of Adv Research in Dynamical & Control Systems,* Vol. 9, No. 8, 2017.

Nathuji, R. and Schwan, K., 2007, October. Virtual power: coordinated power management in virtualized enterprise systems. *In ACM SIGOPS Operating Systems Review,* Vol. 41, No. 6, 265-278.

Lei Xie, Shengbo Chen, Wenfeng Shen and Huaikou Miao (2018) A Novel Self-Adaptive VM Consolidation Strategy Using Dynamic Multi-Thresholds in IaaS Clouds, 10, 52; doi:10.3390/fi10060052

Tsai, J.T., Fang, J.C. and Chou, J.H., 2013. Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *Computers & Operations Research,* 40(12), pp.3045-3055.

Gawali, M.B. and Shinde, S.K., 2018. Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing,* 7(1), p.4.

Cheng, C., Li, J. and Wang, Y., 2015. An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. *Tsinghua Science and Technology,* 20(1), pp.28-39.

Gupta, S., Tiwari, D. and Singh, S., 2015. Energy efficient dynamic threshold based load balancing technique in cloud computing environment. *International Journal of Computer Science and Information Technologies,* 6(2), pp.1023-1026.

Liu X, Zha Y, Yin Q, Peng Y, Qin L (2015) Scheduling parallel jobs with tentative runs and consolidation in the cloud. J Syst Softw 104:141–151.

Yadav, R. and Zhang, W., 2017. MeReg: Managing energy-SLA tradeoff for green mobile cloud computing. *Wireless Communications and Mobile Computing,* 2017.

Kaur, B., Kaur, N. and Singh, R., 2017. A study of energy saving techniques in Green Cloud Computing. *Advances in Computational Sciences and Technology,* 10(5), pp.1191-1197.

Rodriguez MA, Buyya R (2014) Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing* 2(2): 222–235.

Darwish, R.R. and Elewi, A., 2019. A green proactive orchestration architecture for cloud resources. *International Journal of Computers and Applications,* 41(2), pp.112-128.

Zhou, Z., Hu, Z. and Li, K., 2016. Virtual machine placement algorithm for both energy-awareness and SLA violation reduction in cloud data centers. *Scientific Programming,* 2016, p.15.

Buyya, R., Toosi, A. N., & Calheiros, R. N. 2014. Interconnected cloud computing environments: Challenges, taxonomy, and survey. *ACM Computing Surveys,* 47, 1–47.

Received: 07.07.2021 – Accepted: 05.08.2021