

A Novel Energy Efficient Scheduling for VM Consolidation and Migration in Cloud Data Centers

Dasari Yakobu*, Chirra Venkata Rami Reddy, Venkatramaphani Kumar Sistla

Department of CSE, Vignan's Foundation for Science Technology and Research, Vadlamudi 522213, AP, India

Corresponding Author Email: dy_cse@vignan.ac.in

<https://doi.org/10.18280/isi.240512>

Received: 21 May 2019

Accepted: 14 August 2019

Keywords:

virtualization, cloud data center, green computing, energy efficient scheduling algorithm

ABSTRACT

In data centers, the energy-efficient scheduling of virtual machines (VMs) is critical to the full utilization of physical machines (PMs). Considering the sheer amount of data in cloud environment, this paper puts forward a novel energy-efficient scheduling method for VM consolidation and migration in cloud data centers. The proposed method optimizes the energy consumption at cloud data centers through three algorithms: the first algorithm describes the general migration of VMs among PMs; the second algorithm defines the migration of VMs among PMs; the third algorithm explains how the migration takes place. The effectiveness of our method was demonstrated on CloudSim with 5 PMs and 30 VMs, under the constraints of arrival time and deadline. The results show that our method can balance the load of input jobs and schedule the VMs properly, thus reducing the carbon emissions at the cloud data center.

1. INTRODUCTION

Cloud computing is a booming technology in which any one can share and access computing resources such as software, development platforms and infrastructure as a service online with less efforts and less cost. Due to its simplicity and many other attractive features, cloud users are getting increased significantly. It helps any budding organization to manage their product's quality and production without having to spent much of their man power, time and money [1]. It provides both

software and hardware resources online to any user with smart phone, laptop with basic internet connection. The main advantage of cloud technology is that the organization need not to maintain large scale capital expenditures to get access to cloud resources. They can "pay per use" i.e., they need to pay for the amount of resources they used. Cloud is based on the idea of Utility Computing where storage and computing resources are provisioned as metered services similar to public utility services [2]. The Figure 1 shows the Cloud service model.

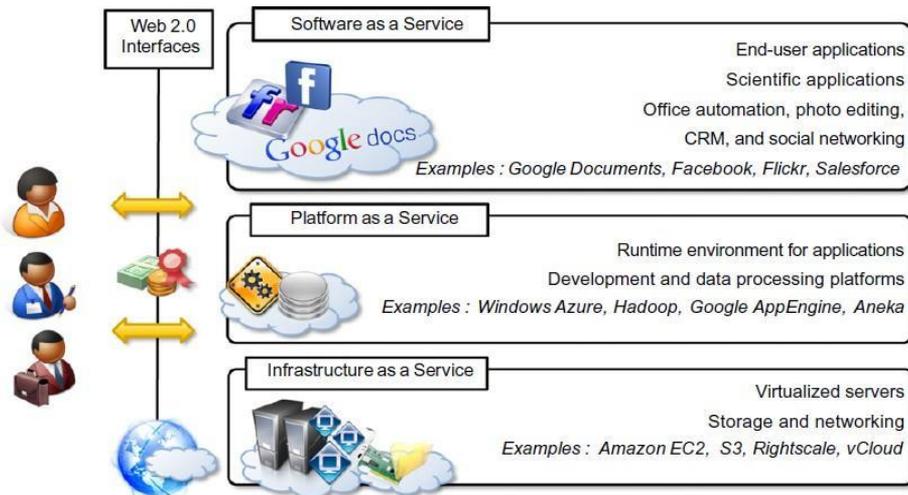


Figure 1. Cloud service models

1.1 Cloud architecture

The architecture of cloud includes the following layers and each layer performs it dedicated responsibilities. The architecture of cloud consists of following layers [2].

Layer0(Network):

Layer1(Hardware): The hardware requirements necessary for cloud

Layer2(Operating System): Any basic operating system that serves as interface between hardware and virtualization layer.

Layer3(Hypervisor): Responsible for virtualization.

Layer4(Middleware): Takes the responsibility of managing cloud, dynamic provisioning of cloud services.

Layer5(Virtual Machines): Infrastructure as a service can be provisioned as virtual machines.

Layer6(Management): It acts in all layers for metering and

billing of cloud services.

Layer7(Security): Functions in all layers to ensure the security of them there by providing full security to cloud.

Layer8(Portal): Amongst all the layers the virtualization layer plays vital role in provisioning the much of cloud services to cloud users dynamically.

Virtualization: Is a technique that through which we can run multiple applications, operating systems on single system with dedicated hardware and software [3-5]. Through virtualization we can creating the virtual version of computing capabilities/resources and share them so that cloud user can interact with those resources. Virtualization provides the creates the logical view of computing resources so that it is possible to run multiple application and operation systems on single OS. This technique is different from multi tasking and reduces the processing overheads. The advantages of virtualization technique include:

- i. Optimum utilization of hardware resources
- ii. Concurrent access of different and conflict applications on same machine
- iii. Isolation and management of services
- iv. Reduction in ownership cost and power consumption

1.2 Cloud characteristics

The basic characteristics of cloud includes [6].

On-demand self service: The computing resources can be provisioned on fly without having human intervention.

Broad network access: Resources can be accessed through internet with basic mechanisms from anywhere to anywhere.

Resource pooling: Resources are pooled from different locations to serve the needs of different customers. It also promotes location independent service pooling.

Rapid elasticity: computing capabilities are provisioned automatically and it can be quickly scaled up and scaled down.

Measured services: Cloud automatically quantifies, monitors, controls, manages and reports the usage of resources to both cloud user and provider for transparency.

1.3 Challenges

Despite of advantages, cloud users are also facing some challenges [7]. Some of them are:

- i. Security and Resource Ownership
- ii. Shared Resources and Software licensing
- iii. Availability/Uptime
- iv. Regulation/Compliance
- v. Product/Service available

In addition to the above mentioned challenges the one more major problem identified with the cloud is energy consumption in data centers. As we know, cloud environment generates huge amount of data which is stored and processed at data center. The data growth in cloud environment is exponential. As we know, data center is composed of set servers and various resources; these should be available to consumer on-demand at any time around the clock.

This continues usage of data center leads to high carbon emission that damages the environment unexpectedly. This is the dark side of the smart cities development.

Cloud providers follows two different virtualization approaches such as full virtualization and para-virtualization. There are four types of virtualizations such as network, storage, server and desktop virtualization.

Hypervisor: It is the key element of virtualization. It is also called as "Virtual Machine Manager" is a layer that allows multiple applications and operating systems to run on single hardware host. Table 1 [2] shows the comparison between different hypervisors developed by Oracle.

In this minor project, we are proposing an energy efficient scheduling algorithm in cloud to reduce the carbon emission at data center. The main objectives of this work are: reducing energy consumption and increase the resource availability. The proposed methodology reduces the carbon emission by load balancing. proper scheduling allows us to improve the resource availability. Resource availability can be determined by calculating the weight of the server using the weight approximation algorithm.

Table 1. Comparison between various Hypervisors

Name	KVM	VirtualBox	VM aware workstation	XEN
Creator	Intel/AMD processor with Qumranet	Innotek	VM ware	Xen source
Host CPU	X86 virtualization, IA64, s390, Power P	x86, x8664, x86, x8664 only	x86, x8664	x86, x8664, IA64
Guest CPU	C, Same as host	Windows, Linux, Mac OS X	x86, x8664	Same as host
License	GPL version 2	GPL version 2	Proprietary Netware, OS/2	GPL
Method of operation	X86 virtualization	X86 virtualization	Para-virtualization	Para-virtualization
Symmetric Multiprocessing on Guest OS	Yes	Yes	Yes	Yes
USB & GUI support	Both	Both	Both	Only GUI
Support Guest OS	No	No	No	Yes,
Live memory drivers allocation	Yes	Yes	Yes	Yes

2. RELATED WORKS

Jiang et al. [8] presented a discussion on how to reduce the Carbon dioxide that is emitted from data centers that were being used continuously for days. So, energy consumption has become a major problem in this area of cloud. This paper talks about EEVS with networking, load balancing and with many more constraints that helps the city to be smarter. Here they use power model that describes the relation between energy

consumption and load balancing [9-10]. EEMCRA algorithm is proposed and used in this paper. EEMCRA is meant for energy efficient minimum criticality routing algorithm. It includes load balancing strategy and routing optimization. To improve the network efficiency, we use E2MR2 algorithm. The main problem is network robustness. To measure robustness, they used and criticality to solve the problem. Graph theory is used to define network topology. Network criticality is inversely proportional to robustness i.e., where the

is more criticality there is less robustness and this parameter is considered in order to solve the problem. We use MCRA algorithm that produces the shortest paths from end to end node [11]. EEMCRA although produces highest energy efficiency it cannot satisfy some constraints like maximum delay and link utilization ratio. To overcome this problem E2MR2 is used where re-routing is considered. Power model uses a function that describes links in between the machines in network which is related to traffic. If the function of link load is zero, it indicates that energy consumption is zero.

Now a day's cloud services are used by many people. So, it could be efficient for cloud users if there is efficient load balancing so that the usage will be increased. This result in low latency and data transmission time decreased. In this paper [12] the load balancing achieved through several protocol mechanisms like HTTP redirection, HTTP requests, DNS responses. They used graphs, Voronoi partitions where the requests are sent to the data center with less distance when one or more data centers can be accessed by a single request of user. These Voronoi partitions also make the set in such a way that it uses less request time, less carbon emission and also minimum electricity cost [13, 14]. But while using Voronoi partition we need to consider the minimum distance between source and destination in order to get better results [15]. This paper also includes the concept of Pair Wise partitioning rule, where two different data centers communicate to each other and then they decide which will be the better route for a particular region. And to study the carbon emission they used to check that amount of carbon released by utilizing EC2 infrastructure. In cooling they suggest to use "free air cooling" where the cooling process is like depending on the temperature it allows the cool air from the environment into the data center to maintain the temperature.

Multi-tenancy mainly works for sharing of resources in a cloud whether it may be public or private. The models used in this paper [16] are Fat tree networking and Hose traffic model. As we know Fat tree is three-tier architecture but for the convenience it is abstracted into two-tier. Edge switches help to resolve the intra-edge traffic while core switches take care of inter-edge traffic. Because of the volatile nature of the traffic in a data center, Hose traffic model is taken into consideration [17]. Here the Virtualization framework has many phases like placement phase, link establishment phase [18, 19]. The main idea of placement phase is to reduce traffic by clearing the inter-edge flows thinking of the future so that cloud resources are used in efficient manner. Second phase is about to select the core switch and establishes link between edge switch and core switch. The strategies used in both the phases are dynamic so that they can be used for further allocation.

Najafi et al. [20] proposed an approach with Harmony Search Algorithm (HSA) to reduce energy consumption in cloud environment. It focuses on reducing virtual machines migration and is suitable for energy efficiency in the infrastructure as a service (IaaS) [21, 22]. This approach primarily identifies best physical system to reallocate the VMs and thereby reduces the energy consumption in cloud environment. The approach works in four steps: Hosts are sorted in descending order by their workloads, VMs with low loads are selected for migrations, and they are sorted in descending order by their migration ranking and are placed on target host by considering threshold value and final step is shutting down the hosts with low load. The proposed approach is compared with PABFD and results shows that proposed

algorithm outperforms than PABFD.

Chen et al. [23] proposed an algorithm (CLB), for reducing energy consumption in cloud environment [24, 25]. This architecture considers server processing power and computer loading. This mechanism follows three scheduling algorithms to reduce energy consumption in cloud. It has five layers. A layer called Cloud Load Balance Monitoring Platform (CLBMP), primarily identified the loads of all the services, detects whether the service is online, and sort the server load and stores it in a database. Another layer called Cloud load balance distribution platform (CLBDP), assign the server based on user request by using wheel load balancing mechanism. Based on user requests cloud server provides storage. This algorithm monitors platforms to obtain the load of each server, computing power and the priority service value (PS). When the cloud server receives a user request, demanded services will enter in to cloud load balancing distribution platform. This platform will get the first half of the service from the database based on their priority service value, and polling method is used to dispatch the services to users.

Li et al. [4] proposed an algorithm EAGLE that balance the utilization of multi-dimensional resources. The algorithm works in such a way that reduces power consumption to the significant level by decreasing the no. of working PMs. The primary objective of this algorithm is, first it selects the best physical machine (PM). Then VMs are identified to deploy in PMs and placed on virtual machine placement (VMP). The number of PMs are minimized to improve the resource utilization [26, 27]. The goal is to reduce the no. of fully loaded resources and their sizes.

Zhou et al. [28] proposed an algorithm, EEOM for optimizing time of triggering and selection of virtual machine. The EEOM considers CPU time and memory factors. It utilizes the Auto-Regression model to prevent the unnecessary VM migrations. The algorithm optimizes time of triggering, selection of virtual machine and location of host machine, taking CPU and memory factors into consideration. It can also predict the future condition of services. This model follows the triggering strategy as follows: if the resource utilization < minimum threshold value then VM is added to the server list else server will be added. It considers the three condition for VM migration. VM's with overloaded CPU, overloaded memory, and minimum load threshold value [29-31].

Zhou et al. [28] proposed a consolidation algorithm named PVDE (prediction-based VM deployment algorithm for energy efficiency) that predicts the load of each server [32, 33]. A linear weighted method is used to predict the load of each server. This algorithm first predicts the load of a server (predictive value of hosts' CPU utilization) using linear weighted method. PVDE sets three thresholds (a,b,c) that results the four categories of data center hosts. i.e., hosts with little, light, optimal and heavy loads. All VMs on little-loaded host are migrated to lightly-loaded hosts. All VMs on optimally-loaded host and lightly-loaded host are kept unchanged. Some VMs on heavily-loaded host are migrated to lightly-loaded hosts. The author consolidated four VM selection algorithms, to reduce the CPU utilization.

Gaochao et al. [34] proposed a load balancing model for the public cloud that follows switch mechanism [35, 36]. For simplifying load balancing process, this model partitions the public cloud in to several parts. The cloud main controller chooses the best partition whenever the service arrives at public cloud. The selection of best partition is done based on its partition status which is categorized in to three types i.e.,

idle, normal, and overload. So, game theory is used in this load balancing strategy in which the normal load status can be viewed as a non-cooperative game.

Zhou et al. [37] proposed an algorithm (TESA) for energy efficiency [30, 31, 38] in cloud data servers. This mainly works on relationship between the energy consumption and (processor) resource utilization. In this algorithm, VMs migration takes place when VMs on host are highly loaded and some on host are lightly loaded. These VMs are migrated to another host with proper load. To avoid excessive migration, all the VMs on properly loaded host or middling loaded host are kept unchanged. The author had proposed five selection polices to select the VMs for migration. The main goal of selection polices is that select the VMs for migration in such a way that all the host's load is balanced and reduce the CPU utilization.

3. PROPOSED METHODOLOGY FOR VM CONSOLIDATION, MIGRATION AND SCHEDULING

Although cloud computing has got huge attention from the industry, still it is suffering from various issues like security, service availability, QoS, standardization, and power consumption [39]. Research studies indicated that efficient energy management in cloud data center to build green computing is most burning problem. It also indicated that over 90% of a data farm's power is devoured by the IT equipment [40, 41]. Hence cloud infrastructure providers need to adopt measures to ensure the energy saving [39, 42, 43]. We studied the state-of-the art techniques for power saving in the IaaS level [44]. The studies indicated that through effective use of cloud core technology virtualization, we can reduce power consumption in cloud. In our work, using virtualization [3], we proposed a methodology, "a novel energy efficient scheduling for VM consolidation and migration in cloud data centers", for building of green cloud [21, 42, 45].

The proposed methodology ensures the optimized energy consumption in a systematic procedure with three algorithms (Figure 2). Each algorithm has a defined task in which Algorithm-1 describes the general migration of VMs among PMs. Algorithm-2 decides the migration of VMs among PMs. Algorithm-3 explains how the migration takes place.

As we know that cloud environment generates huge amount of data, this data can be stored and processed at data center, where data growth is exponential. Cloud data centers are composed of set of servers and other resources which must be running to provide the consumers requests at any time round the clock.

This continuous usage of data centers leads to huge amount of carbon emissions which is a dark side of cloud technology. This is where virtualization comes into picture. Virtual machine scheduling is one of the important factors that control the energy consumption in the cloud environment. The scheduling of VMs among PMs is done by ensuring that each PM is fully loaded. This mechanism could reduce the power consumption by putting some PMs in sleep state. The main idea of proposed methodology is to only keep that PMs active that have at least one core in active state and turn off other PMs. The proposed methodology works at three levels, at the first level (Algorithm: 1), it describes the general scheduling VMs to PMs, at the second level(Algorithm 2), it decides whether the scheduling should takes place or not and at the third level (Algorithm: 3), it explain how scheduling takes place if it is mandatory.

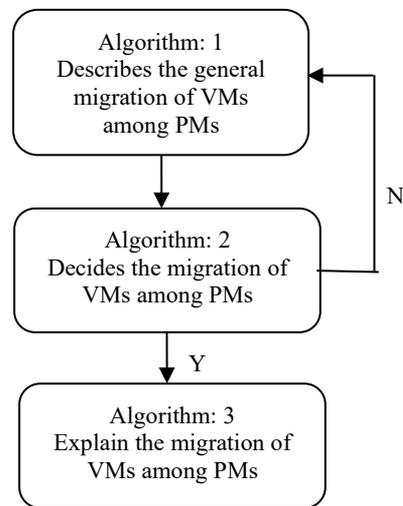


Figure 2. Proposed methodology

Level-1: The algorithm 1 describes the general phenomenon of scheduling VMs among various PMs.

In cloud VMs arrive at an increasing rate and it can be defined by its computations, arrival time, start time and deadline. VM should be assigned to PM whenever it arrives. The assignment of VM to PMs is done on the basis of Optimal Frequency (OF) of each PM. Optimal Performance Power Ratio (OPPR) is computed based on the Optimal Frequency and physical machines with higher ratio will be allocated VMs prior to other PMs. And the algorithm-1 is as follows.

Algorithm:1

Inputs: List of PMs, List of VMs

Output: Scheduling of VMs

1. Compute the Optimal Performance Power Ratio of all the Physical Machines in the list and sort the list in the decreasing order of ratios.

2. For every time period (t),
Check the VM List for VMs machines to be allocated in that time period.

For every VM to be allocated in (t) Check for idle cores in PM_List

If found:

Allocate the VM to that core. Start the VM execution and set VM.start_time = t
Add (VM,PM,Core) to Allocation List

Else:

Check in the next PM and continue until the VM is allocated or no more PMs are left.

If a VM is not allocated successfully:

Update the required resource for VM (VM.rr)
Add the VM to the head of VM List to allocate in the next time Period.

3. After allocating VMs in (t), set the Optimal Frequency for each active core by considering the number of active cores in each PM.

For every (VM,PM,Core) in Allocation List:

If VM has completed its execution:

Make the core inactive

If all the cores in the PM are inactive:

Set PM to sleep state.

Remove (VM,PM,Core) from Allocation List.

4. For each (VM,PM,Core) in Allocation List: Check for PMs with higher OPPR than PM.

If found:

Check if any of its cores are idle or has enough resource for VM

If Found:

Migrate the VM to the new PM and Core.

Remove (VM,PM,Core) from Allocation List.

Add (VM,PM', Core') to Allocation List.

5. Repeat from Step 2 Until all the VMs are either done or failed.

The PPR of each PM can be computed by using the following equation.

$$PPR = \frac{Freq_i * \frac{(AR-ALR)}{AR} * No.ofCores}{\sum_{i=0}^n PUE} \quad (1)$$

where, $Freq_i$ is a frequency of PM, AR is available resource, ALR is allocated resource, PUE is a power usage effectiveness.

To compute the central tendency among the PPRs of each PMs, variance is being computed using Eq. 2.

$$v = \sqrt{\frac{(o-\mu)^2}{n-1}} \quad (2)$$

where, μ is the average of PPR values of various PMs. o is the ppr value of the individual PM and n is the total no. of machines. The need of the scheduling is calculated based on the value of v .

$$v = \begin{cases} Yes, & v < \Delta \\ No, & otherwise \end{cases} \quad (3)$$

Level-2: Algorithm-2 considers the Variance of OPPRs of all PMs for better assignment of VMs to PMs. i.e. The algorithm first sets the threshold value (tolerance) for all OPPRs of all the PMs. Then it computes the Variance of all the OPPRs and compares it with tolerance. If variance $>$ tolerance, it concludes that all the PMs are equally loaded and no scheduling is required. Otherwise it defines that some of the machines are heavy loaded and the scheduling is required. In this case the machines which were lightly loaded will complete the task and enters into sleep mode in the short time. The assignment of VMs to PMs is done by calling algorithm-3. The algorithm-2 works as follows.

Algorithm:2

Input: OPPRs of all PMs

Output: Return True/False

1. Determine the mean of OPPRs of all PMs

$\mu = \text{mean of } O_i = \{O_1, O_2, O_3, O_4, O_5\}$ of all PMs $P = \{P_1, P_2, P_3, P_4, P_5\}$

$\mu = \sum_{i=1}^n O_i / n$

2. Determine the Variance of all OPPRs of all PMs

3. If Δ is tolerance of OPPR values of all PMs which controls the scheduling of VMs among the PMs if $v < \Delta$ then

all the $P = \{p_1, p_2, p_3, p_4, p_5\}$ are equally weighted and there is no need of VM migration

Else:

Migrate the VM among PMs by calling algorithm3

Level-3: Once after algorithm-2 decides the scheduling of

VMs among PMs, algorithm-3 takes OPPRs of all the PMs and assigns five bins (Bin1: min, Bin2: 25%, Bin3: median, Bin4: 75%, Bin5: max) to each OPPR. Then it compares the OPPRs of each PM with bin values. If any PM's OPPR lies between 25% and median, then assign the its VM to PM of Bin3. If any PM's OPPR lies between min and 25%, then it assigns the VM to PM of Bin4. If any PM's OPPR lies between median and 75%, then it assigns the VM to PM of Bin2. the algorithm-3 works as follows.

Algorithm:3

Input: OPPRs of all PMs

Output: Assignment of VM to corresponding PM

1. Categorize the OPPR values in to five bins. ie Bin:1 min; Bin2: 25%; Bin3: median; Bin4: 75%; Bin5: max

2. *if median < OPPR < 75% then*

assign VM to PM of Bin2

if 25% < OPPR < median then

assign VM to PM of Bin3

if min < OPPR < 25% then

assign VM to PM of Bin4

4. RESULTS & DISCUSSIONS

We demonstrated the scheduling of proposed methodology on CloudSim with five Physical Machines and thirty Virtual Machines with its arrival time and deadline constraints. Time period is set as 1 second, VMs are assigned to PMs in each time frame. VM which is not assigned in the given time frame can be considered first in the next time frame. The machines with parameters are given in the below table 2.

Table 2. Physical machines with different parameters

No. of cores	No. of states	Set of Frequencies (Hz)	CPU Power (W)	Idle Power (W)	Peak Power (W)
4	4	[2.4, 2.7, 3.0, 3.2]	63	47	148
2	4	[1.2, 1.5, 1.8, 2.2]	65	32	101
6	4	[1.5, 1.8, 2.3, 2.9]	73	55	172
4	4	[2.3, 2.7, 2.9, 3.2]	67	55	173
2	4	[1.5, 1.7, 2.3, 2.7]	79	51	159

Below Figure 3 shows that initially there is no VM left for the assignment.

```
-----Time period (1):-----
2018-11-18 21:08:05.672998 to 2018-11-18 21:08:06.672998
VMs left to be allocated : VM(t) : []
-----End of Time Period-----
```

Figure 3. No VMs for assignment

Figure 4 shows assignment of VM4 to Core0 of PM3 and initiates after assignment.

```
-----Time period (2):-----
2018-11-18 21:08:06.674011 to 2018-11-18 21:08:07.674011
VMs left to be allocated : VM(t) : [4]
VM 4 allocated to PM 3 Core [0]
Start time of VM 4 : 2018-11-18 21:08:07.674011
-----End of Time Period-----
```

Figure 4. Assignment of VM4 to Core0 of PM3

Figure 5 shows three VMs, VM1, VM5 and VM6 left for assignment. It also shows the assignment of VM1 to PM3 core1, VM5 to PM3 core2 and VM6 to PM3 core3 and their start times.

```
-----Time period (3):-----
2018-11-18 21:08:07.677845 to 2018-11-18 21:08:08.677845
VMs left to be allocated : VM(t) : [1, 5, 6]
VM 1 allocated to PM 3 Core [1]
Start time of VM 1 : 2018-11-18 21:08:08.677845
VM 5 allocated to PM 3 Core [2]
Start time of VM 5 : 2018-11-18 21:08:08.677845
VM 6 allocated to PM 3 Core [3]
Start time of VM 6 : 2018-11-18 21:08:08.677845
-----End of Time Period-----
```

Figure 5. Assignment of VM1, VM5 and VM6 to various cores of PM3

```
-----Time period (4):-----
2018-11-18 21:08:08.680084 to 2018-11-18 21:08:09.680084
VM 4 at PM 3 Core [0] has completed its execution.
VMs left to be allocated : VM(t) : [2, 3, 9, 26]
VM 2 allocated to PM 3 Core [0]
Start time of VM 2 : 2018-11-18 21:08:09.681089
VM 3 failed !! (Enough resource not available)...
VM 9 allocated to PM 3 Core [4]
Start time of VM 9 : 2018-11-18 21:08:09.681089
VM 26 allocated to PM 3 Core [5]
Start time of VM 26 : 2018-11-18 21:08:09.681089
VM 1 has been reallocated to PM 4 Core [0]
VM 6 has been reallocated to PM 4 Core [1]
VM 9 has been reallocated to PM 4 Core [2]
-----End of Time Period-----
```

Figure 6. Assignment of VM1, VM6 and VM9 to various cores of PM4

Figure 6 shows assignment of VM2, VM9 and VM26 to PM3 core0, core4 and core5 and their start times. It also shows the failure of VM3 assignment due to inadequate resources.

Figure 7 shows the Re-assignment of VM15, VM16 and VM27 to Various cores of PM4. It also shows the completion of VM12, VM17, VM18 and VM28 execution at PM2 & PM4.

```
-----Time period (12):-----
2018-11-18 21:08:16.711542 to 2018-11-18 21:08:17.711542
VM 12 at PM 4 Core [0] has completed its execution.
VM 18 at PM 4 Core [2] has completed its execution.
VM 17 at PM 4 Core [3] has completed its execution.
VM 28 at PM 2 Core [1] has completed its execution.
VMs left to be allocated : VM(t) : []
VM 15 has been reallocated to PM 4 Core [0]
VM 16 has been reallocated to PM 4 Core [2]
VM 27 has been reallocated to PM 4 Core [3]
-----End of Time Period-----
```

Figure 7. Re-assignment of VM15, VM16 and VM27 to Various cores of PM4

Below Figure 8 shows the no. of failure VMs assignment.

```
No. of failed VM's :1
```

Figure 8. Status of failed VM assignment

```
Execution time of Cores :
PM 4 Core (0,) executed for 7.091796874999999 seconds.
PM 4 Core (1,) executed for 0.90234375 seconds.
PM 4 Core (2,) executed for 7.654947916666666 seconds.
PM 4 Core (3,) executed for 4.791666666666666 seconds.
PM 1 Core (0,) executed for 1.3888888888888888 seconds.
PM 1 Core (1,) executed for 0.0 seconds.
PM 3 Core (0,) executed for 7.005747126436782 seconds.
PM 3 Core (1,) executed for 1.2608695652173914 seconds.
PM 3 Core (2,) executed for 6.973832528180354 seconds.
PM 3 Core (3,) executed for 1.8260869565217392 seconds.
PM 3 Core (4,) executed for 2.5555555555555554 seconds.
PM 3 Core (5,) executed for 7.907833120476798 seconds.
PM 2 Core (0,) executed for 4.624446457326892 seconds.
PM 2 Core (1,) executed for 3.902611496184275 seconds.
PM 2 Core (2,) executed for 2.6521739130434785 seconds.
PM 2 Core (3,) executed for 2.0869565217391304 seconds.
PM 5 Core (0,) executed for 0.0 seconds.
PM 5 Core (1,) executed for 0.0 seconds.
```

Figure 9. Execution status of various cores of PMs

Figure 9 above shows the execution status of all VMs at various PMs along with their execution time.

The performance evaluation of 30 VMs over 5 PMs is carried on *CloudSim* tool. In the process of evaluation, the following parameters have been considering which includes, arrival times, deadlines and required resources of VMs, performance power ratios, power consumed by PM.

Arrival time and deadlines (ranges from 21.08.09 to 21.08.24) of VMs have been shown in Figure 10, Required resources frequency (Hz) of VMs have been shown in Figure 11.

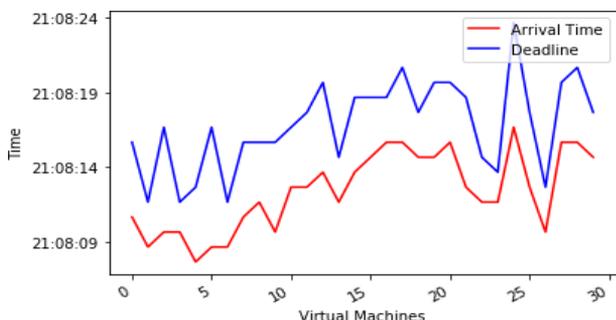


Figure 10. VMs Vs time

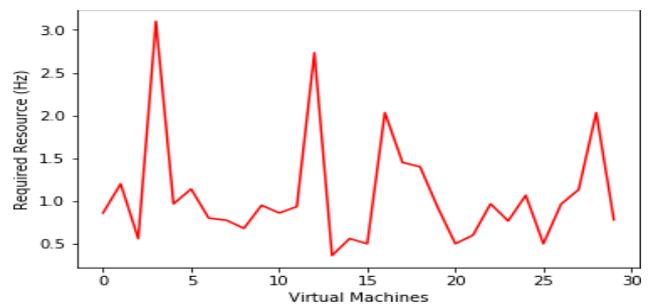


Figure 11. VMs Vs required resources

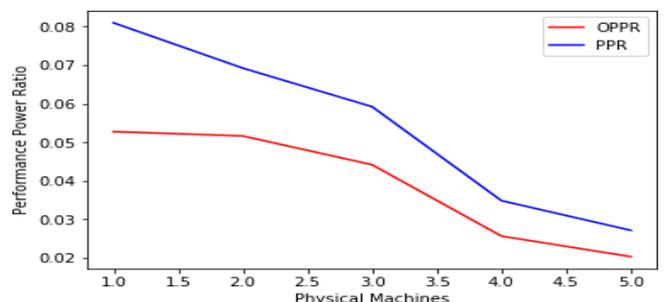


Figure 12. PMs Vs performance power ration

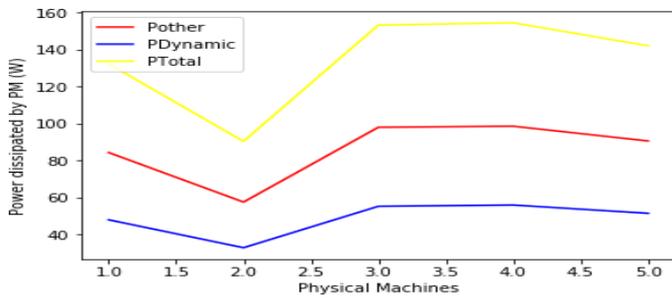


Figure 13. PMs Vs power requirement

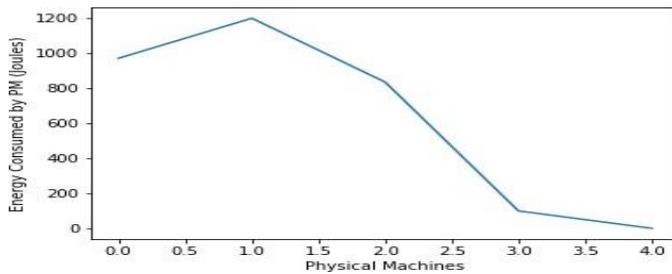


Figure 14. PMs Vs power consumption

Performance power ratios and power dispatched by PMs have been shown in Figure 12 & Figure 13, respectively. Finally Figure 14 shows after applying proposed methodology, gradual decrease in power consumption by putting one PM in sleep state.

5. CONCLUSION

Rigorous review has been done cloud core concepts and made an extensive study on available energy efficient techniques to reduce energy consumption in cloud for building the green cloud computing systems. In this paper, authors have proposed and implemented a methodology that is not only helps us to strong cloud but also helps to reduce CO2 emission from a cloud data center. The proposed methodology has been implemented and tested on *CloudSim* with five physical machines and 30 virtual machines. Experimental results show that proposed methodology could reduce the power consumption in cloud data centers.

REFERENCES

- [1] Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., Merle, P. (2018). Elasticity in cloud computing: State of the art and research challenges. *IEEE Transactions on Services Computing*, 11(2): 430-447. <https://doi.org/10.1109/TSC.2017.2711009>
- [2] Buyya, R., Vecchiola, C., ThamaraiSelvi, S. (2013). *Mastering the cloud-text book*. MK Morgan Kaufmann, Elsevier Inc.
- [3] Shoaib, Y., Das, O. (2014). Pouring cloud virtualization security inside out. *Cryptography and Security*, pp. 1-13.
- [4] Li, X., Qian, Z., Lu, S., Wu, J. (2013). Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Math. Comput. Model.*, 58(5-6): 1222-1235. <https://doi.org/10.1016/j.mcm.2013.02.003>
- [5] Price, M. (2014). The paradox of security in virtual environments. *Computer (Long Beach, Calif.)*, 41(11): 22-28. <https://doi.org/10.1109/MC.2008.472>
- [6] Avram, M.G. (2014). Advantages and challenges of adopting cloud computing from an enterprise perspective. *Procedia Technol.*, 12: 529-534. <https://doi.org/10.1016/j.protecy.2013.12.525>
- [7] La Porta, T.F. (2002). Introduction to the IEEE transactions on mobile computing. *IEEE Trans. Mob. Comput.*, 1(1): 2-3. <https://doi.org/10.1109/TMC.2002.1011055>
- [8] Jiang, D., Zhang, P., Lv, Z., Song, H. (2016). Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. *IEEE Internet Things J.*, 3(6): 1437-1447. <https://doi.org/10.1109/JIOT.2016.2613111>
- [9] Addis, B., Ardagna, D., Capone, A., Carello, G. (2014). Energy-aware joint management of networks and Cloud infrastructures. *Comput. Networks*, 70: 75-95. <https://doi.org/10.1016/j.comnet.2014.04.011>
- [10] Alvi, S.A., Shah, G.A., Mahmood, W. (2015). Energy efficient green routing protocol for Internet of Multimedia Things. 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, Singapore. <https://doi.org/10.1109/ISSNIP.2015.7106958>
- [11] Chen, J., Gong, Y., Fiorani, M., Aleksic, S. (2015). Optical interconnects at the top of the rack for energy-efficient data centers. *IEEE Commun. Mag.*, 53(8): 140-148. <https://doi.org/10.1109/MCOM.2015.7180521>
- [12] Doyle, J., Shorten, R., O'mahony, D. (2013). Stratus: Load balancing the cloud for carbon emissions control. *IEEE Trans. Cloud Comput.*, 1(1): 116-128. <https://doi.org/10.1109/TCC.2013.4>
- [13] Liu, Z. (2011). Greening geographical load balancing thesis. Master's thesis, Calif. Inst. Technol., 2011(1): 1-54.
- [14] Pathan, M., Vecchiola, C., Buyya, R. (2008). Load and proximity aware request-redirection for dynamic load distribution in peering CDNs. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 5331: 62-81. https://doi.org/10.1007/978-3-540-88871-0_8
- [15] Fridleifsson, I.B., Bertani, R., Huenges, E., Lund, J., Ragnarsson, A., Rybach, L. (2008). The possible role and contribution of geothermal energy to the mitigation of climate change. *IPCC Scoping Meet. Renew. Energy Sources*, pp. 59-80.
- [16] Duan, J., Yang, Y. (2017). A load balancing and multi-tenancy oriented data center virtualization framework. *IEEE Trans. Parallel Distrib. Syst.*, 28(8): 2131-2144. <https://doi.org/10.1109/TPDS.2017.2657633>
- [17] Mysore, R.N., Porter, G., Vahdat, A. (2013). FasTrak: Enabling express lanes in multi-tenant data centers. *Conex. 2013 - Proc. 2013 ACM Int. Conf. Emerg. Netw. Exp. Technol.*, pp. 139-150. 10.1145/2535372.2535386
- [18] Mijumbi, R., Serrat, J., Gorricho, J.L., Bouten, N., De Turck, F., Boutaba, R. (2016). Network function virtualization: State-of-the-art and research challenges. *IEEE Commun. Surv. Tutorials*, 18(1): 236-262. <https://doi.org/10.1109/COMST.2015.2477041>
- [19] Kansal, N.J., Chana, I. (2012). Existing load balancing techniques in cloud computing: A systematic review. *J. Inf. Syst. Commun.*, 3(1): 87-91.
- [20] Najafi, M., Mohebbi, K. (2016). An approach to reduce

- energy consumption in cloud data centers using harmony search algorithm. *Int. J. Cloud Comput. Serv. Archit.*, 6(4): 1-10. <https://doi.org/10.5121/ijccsa.2016.6401>
- [21] Atrey, A., Jain, N., Iyengar, N.C.S. (2014). A study on green cloud computing. *Int. J. Grid Distrib. Comput.*, 6(6): 93-102. <https://doi.org/10.14257/ijgdc.2013.6.6.08>
- [22] Beloglazov, A., Abawajy, J., Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Futur. Gener. Comput. Syst.*, 28(5): 755-768. <https://doi.org/10.1016/j.future.2011.04.017>
- [23] Chen, S.L., Chen, Y.Y., Kuo, S.H. (2017). CLB: A novel load balancing architecture and algorithm for cloud services. *Comput. Electr. Eng.*, 58: 154-160. <https://doi.org/10.1016/j.compeleceng.2016.01.029>
- [24] Bryhni, H., Klovning, E., Kure, Ø. (2000). Comparison of load balancing techniques for scalable Web servers. *IEEE Netw.*, 14(4): 58-64. <https://doi.org/10.1109/65.855480>
- [25] Buyya, R. (2009). Cloud Analyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments. *Distrib. Comput. Proj. Csse Dept., Univ. Melb.*, pp. 433-659.
- [26] Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N., Jiang, G. (2009). Power and performance management of virtualized computing environments via look ahead control. *Cluster Comput.*, 12(1): 1-15. <https://doi.org/10.1007/s10586-008-0070-y>
- [27] Wang, M., Meng, X., Zhang, L. (2011). Consolidating virtual machines with dynamic bandwidth demand in data centers. *Proc. - IEEE INFOCOM*, pp. 71-75. <https://doi.org/10.1109/INFOCOM.2011.5935254>
- [28] Zhou, Z., Hu, Z.G., Yu, J.Y., Abawajy, J., Chowdhury, M. (2017). Energy-efficient virtual machine consolidation algorithm in cloud data centers. *J. Cent. South Univ.*, 24(10): 2331-2341. <https://doi.org/10.1007/s11771-017-3645-z>
- [29] Li, J., Peng, J., Lei, Z., Zhang, W. (2011). An energy-efficient scheduling approach based on private clouds. *J. Inf. Comput. Sci.*, 8(4): 716-724.
- [30] Sindhu, S., Mukherjee, S. (2011). Efficient Task Scheduling Algorithms for Cloud Computing Environment. *HPAGC 2011: High Performance Architecture and Grid Computing*, Chandigarh, India, pp. 79-83. https://doi.org/10.1007/978-3-642-22577-2_11
- [31] Weng, C., Wang, Z., Li, M., Lu, X. (2009). The hybrid scheduling framework for virtual machine systems. *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pp. 111-120. <https://doi.org/10.1145/1508293.1508309>
- [32] Clark, C., Fraser, K., Hand, S., Hansen, J.G. (2005). Live migration of virtual machines. *NSDI'05 Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, pp. 273-286.
- [33] Magalhães, D., Calheiros, R.N., Buyya, R., Gomes, D.G. (2015). Workload modeling for resource usage analysis and simulation in cloud computing. *Comput. Electr. Eng.*, 47: 69-81. <https://doi.org/10.1016/j.compeleceng.2015.08.016>
- [34] Xu, G., Pang, J., Fu, X. (2013). A load balancing model based on cloud partitioning for the public cloud. *Tsinghua Sci. Technol.*, 18(1): 34-39. <https://doi.org/10.1109/TST.2013.6449405>
- [35] Chaczko, Z., Mahadevan, V., Aslanzadeh, S., Mcdermid, C. (2011). Availability and load balancing in cloud computing. *International Conference on Computer and Software Modeling, IPCSIT vol.14 (2011)*, IACSIT Press, Singapore.
- [36] Goudarzi, Z. (2017). Effective load balancing in cloud computing. *Int. J. Intell. Inf. Syst.*, 3(6): 1. <https://doi.org/10.11648/j.ijis.s.2014030601.11>
- [37] Zhou, Z., Hu, Z.G., Song, T., Yu, J.Y. (2015). A novel virtual machine deployment algorithm with energy efficiency in cloud computing. *J. Cent. South Univ.*, 22(3): 974-983. <https://doi.org/10.1007/s11771-015-2608-5>
- [38] Deore, S., Patil, A.N., Bhargava, R. (2012). Energy-efficient scheduling scheme for virtual machines in cloud computing. *Int. J. Comput. Appl.*, 56(10): 19-25. <https://doi.org/10.5120/8926-2999>
- [39] Jing, S.Y., Ali, S., She, K., Zhong, Y. (2013). State-of-the-art research study for green cloud computing. *J. Supercomput.*, 65(1): 445-468. <https://doi.org/10.1007/s11227-011-0722-1>
- [40] Barth, S. (2010). Reducing data center power consumption. *KM World*, 19(7): 8-20.
- [41] Sawyer, R.L. (2011). Calculating total power requirements for data centers. *Schneider Electr. - Data Cent. Sci. Cent.*, pp. 1-10.
- [42] Faucheux, S., Nicolai, I. (2011). IT for green and green IT: A proposed typology of eco-innovation. *Ecol. Econ.*, 70(11): 2020-2027. <https://doi.org/10.1016/j.ecolecon.2011.05.019>
- [43] Joumaa, C., Kadry, S. (2011). Green IT: Case studies. *Energy Procedia*, 16: 1052-1058. <https://doi.org/10.1016/j.egypro.2012.01.168>
- [44] Aslam, S., Shah, M.A. (2016). Load balancing algorithms in cloud computing: A survey of modern techniques. *2015 Natl. Softw. Eng. Conf. (NSEC)*, Rawalpindi, Pakistan, pp. 30-35. <https://doi.org/10.1109/NSEC.2015.7396341>
- [45] White, D.J., Schmidt, D.C. (2012). Model-driven auto-scaling of green cloud computing infrastructure. *Futur. Gener. Comput. Syst.*, 28(2): 371-378. <https://doi.org/10.1016/j.future.2011.05.009>