International Journal of Engineering & Technology, 7 (3.6) (2018) 179-186

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Dynamic VM Consolidation Enhancement for Designing and Evaluation of Energy Efficiency in Green Data Centers Using Regression Analysis

A.V. Sajitha[1*], A.C. Subhajini[2]

[1]Research Scholar, Department Of Computer Applications, Noorul Islam Centre For Higher Education, Tamilnadu, India.
[2]Assistant Professor, Department Of Computer Applications, Noorul Islam Centre For Higher Education, Tamilnadu, India.
E-Mail:Acsubhajini@Yahoo.Com
**Corresponding Author E-Mail:Sajithaav09@Gmail.Com*

Abstract

Enhancement of dynamic Virtual Machines (VM) consolidation is an efficient means to improve the energy efficiency via effective resources utilization in Cloud data centers. In this paper, we propose an algorithm, Energy Conscious Greeny Cloud Dynamic Algorithm, which considers multiple factors such as CPU, memory and bandwidth utilization of the node for empowering VM consolidation by using regression analysis model. This algorithm is the combination of several adaptive algorithms such as EnCoReAn (UPReAn) for Predicting the Utility of a host), Overload and Under-load detection), VM Selection and Allocation algorithms, which helps to achieve live VM migration by switching-off unused servers to low-power mode (i.e., sleep or hibernation), thus saves energy and efficient resource utilization. This approach reduces the operational cost, computation time and increase the scalability. The experimental result proves that, the proposed algorithm attains significant percentage in reduction of energy consumption rather than existing VM consolidation strategies.

Keywords: *Cloud computing, green cloud computing, data center, virtual machine placement, dynamic VM consolidation, VM live migration, linear regression.*

1. Introduction

Cloud computing is a emerging and promising computing architecture with exciting characteristics like on-demand self-service, dynamic scaling, metered services etc. that are to be accomplished over a distributed network in which enormous resources are hoarded in large data centres and which can be accessed in economic, convenient and effective way through Internet based computing for its vast number of users. In early days, users whether an individual or an organization is use their own resources – such as application, storage and network by themselves. Now-a-days, IT has completely grown with excellent supremacy and obtained a dignified position for itself and all those that associated with it. At the present circumstances, resources provided by cloud allow users to get on demand access with least managing attempt based on their requirements in a metered basis [1].

Cloud architecture consists of three layers which are providing various types of services. Software as a Service (SaaS) helps consumers to access software applications online as per on demand, Platform as a Service (PaaS) tenders computing resources by means of a platform whereupon applications as well as services can be build up, where as Infrastructure as a Service (IaaS) conveys fundamental infrastructure supporting services. As the burgeoning needs of consumers for these computing services, the cloud service providers such as Google, Microsoft, Yahoo, Apple, Amazon etc. are encouraging to deploy large amount of power consuming data centres which make adverse effect to our planet. Each and every data centre is an integrated repository consists of thousands of physical machines arranged in hundreds of racks that can be operated millions of Virtual Machines. It could consume as much power as a small hydroelectric power station could produce [2]. The environmental impact is the CO_2 emission, which makes up lion share of the Green House Gas (GHG) into the atmosphere. It will create Global Warming which is the main factor of the increment of Earth's average surface temperature.

The amount of energy consumption of data centers is replicating every five years. as stated by the Natural Resources Defense Council (NRDC), on a national scale, data centers as a whole consumed 92 billion kilowatt-hours (kWh) of electric-power in 2014, and they will be consumed 139 billion kWh by the year 2020.Presently, data centers consuming about 3% of the entire global electricity production as a result these are emitting 200 million metric tons of CO_2. In accordance with the report of European Union, decrease in CO_2 emission volume of 15%-30% is required, by the year 2020 to maintain the global temperature increase below 2°C, a [3].

Green computing or green technology specifies to the eco-friendly utilization of computers and any other technology-related resources. As a result of such potential impacts to the atmosphere, the green cloud computing initiative has emerged as part of the green IT vision. Green Cloud Computing is envisaged to attain not only efficient handling and proper utilization of computational infrastructure, but also diminishing the energy consumption. It offers a simulation environment for energy-aware cloud data centers [4].

Cloud computing leverages virtualization technology for efficient provisioning of resources. The Infrastructure as a Service layer can be used for virtualization capabilities to enhance the accessibilities of Cloud Data Center (CDC) infrastructure. Virtualization technology allows the creation of VMs. A virtual machine (VM) is an operating system or application environment that is installed on software, which mimics the committed equipment. The naive user has the indistinguishable experience on a virtual machine like they execute programs as real physical machines.

Virtualization eliminates energy inefficiency by placing several VM in a single physical server through live VM migration techniques. VM consolidation is a technique to lessen the number of active PMs by migrating and consolidating the VMs into reduced number of physical machines [5]. VM consolidation includes VM placement (the process of selecting the appropriate host for the given VM and VM Migration (is the task of moving a VM from one physical hardware environment to a different one).

In this paper, we used to consider the live VM migration which is the movement of a operating VM from one physical host to other one without disconnecting its client, storage, network connection, memory of the VMs that are accessed from its original host [6]. Live Migration is carry out for achieving energy efficiency, high availability of physical servers and load balancing in Cloud Data Centers. Here, when no VMs are provisioning, if the host is idle, and then it is switched off completely or powered in minimum power consumption mode (sleep or hibernation). In this work, we are proposing an algorithm, Energy Conscious Greeny Cloud Dynamic Algorithm which helps dynamic VM consolidation to live VM migration that is switch-off unused servers or put it to sleep or hibernation mode which saves energy and efficient resource utilization. Dynamic VM consolidation approaches persuade dynamic features of Cloud model, both Physical Machines and their respective Virtual Machines are recurrently monitored. To assist the reduction the number of active PMs and promotion of the quality of services delivered, on every occasion a PM becomes a hot or cold spot, its VMs are reassigned in an optimal machine by live VM migration [7].

The article categorizes as follows: Section II reviews related works. Section III explains the system architecture presented in this research. Section IV discusses the proposed VM consolidation mechanism. Section V analyses a performance appraisal of the proposed method. Finally, a conclusion and future enhancements are presented in section VI.

2. Literature Review

In this segment, we present pertinent approaches recommended in the literature for accomplishing energy efficiency in CDCs. In the field of cloud computing, energy management guidelines and VM consolidation techniques are playing a key role of achieving energy efficiency. Investigated the solution for bi-objective optimization problem such as to limit the VM migration cost to save energy by way of dynamic VM consolidation in a heterogeneous cloud datacenter. A migration cost estimation method and upper bound estimation method for saving maximum energy, a consolidation score function is formulated [8].

Task scheduling algorithm (MinDelay) that accommodates service request and Energy Conscious Algorithm (ECTC) that minimizes energy consumption in a cloud system in a live VM migration process [9]. A joined energy efficient, forecast based VM placement and migration prototype for resource allotment. They proved that the framework diminishes the performance degradation due to overloads of host by reducing the number of PMs which resulted a significant reduction of energy consumption [10]. VM consolidation framework named M-convex optimization framework for an automated VM consolidation process for assigning VMs and Servers with minimum system reconfiguration. Through this approach performance efficiency of data centres is achieved and scalability is improved [11].

Utilization Prediction aware VM Consolidation (UP-VMC) - a dynamic VM consolidation approach which constructs a VM consolidation as a means of multi-objective vector bin packing dilemma. In order to consolidate VMs into the shortened number of dynamic PMs, it takes both the present and future resources utilization. A regression based forecasting model is used for the prediction of forthcoming resource utilization. A VM allocation algorithm selects a PM enriched with an adequate amount of resources currently and which are used to reallocating the VM in future. They implemented this work in single tier cloud environment architecture. Hence scalability could not be guaranteed [12]. Managing the workload of VM objects are provided by large enterprise clouds by a fully de-centralized manner in scalable and energy-efficient method. In this approach, the multiple resources of the data center are efficiently arranged into a hypercube configuration via exercising a number of distributed load balancing algorithms[13].

Energy-efficient and SLA-aware dynamic VM consolidation mechanism (PCM) algorithm which combined of four algorithms: Over loading host detection algorithm, Under-loading host detection VM Selection algorithm, and VM Placement algorithm. In their approach they consider multiple parameters for predicting the future host utilization as RAM, CPU and network Bandwidth. They proved the efficiency of their algorithm through simulation with other four benchmarking algorithms by improving energy efficiency and avoiding SLA violation [14].

Dynamic Overbooking algorithm which collectively controls virtualization potentials and Software Defined Networking (SDN) for VM with traffic consolidation. Through Network overbooking facility, SDN can combine network traffic and manage Quality of Service (QoS) dynamically. In this approach they estimated resource allocation ratio according to the past data monitoring from the online investigation of the host and network consumption with no prior-knowledge about the workloads. They can be achieved energy efficiency and a huge energy cost savings by minimizing the misuse of over provisioned resources, then together reduced the SLA violation by distributing adequate resources for the real workload [15].

Two effective and efficient Virtual Data Centre Embedding (VDCE) approaches, NSS-JointSL and NSS-GBFS, to deal VDCE issue with the aim of reducing energy usage in meager-workload Cloud Data Centres (CDCs) by reducing embedding charge in hefty-workload CDCs. But the challenge of their work is much more computation time while embedding a VDC request, when more types of physical resources are taken into account [16].

VM Consolidation algorithm with multiple Usage Prediction (VMCUP-M) for enhancing energy effectiveness in cloud data centres. It considers the various resource types and its estimated utilization. While using the actual and predicted utilization, it is easy to identify the overloaded and under loaded hosts in data centres. This algorithm proves better in reduction of workload in individual host and providing energy efficiency while preserving SLA [17].

3. Proposed Work

By means of VM consolidation, cloud suppliers can be made eminent their resource deployment whilst cutting down the energy consumption of cloud data centres. The dynamic VM consolidation is a four level process:

- Identification of a host is overloaded or not for one or more VMs are to be reallocated to another host to cut down the host utilization.
- Identification of a host is under-loaded or not for reallocating the VMs to other host to switch off the same to save energy by wasting it in idle state.

- Selection of most suitable VMs for live migration from overload and under-loaded host to ensure energy efficiency, avoidance of SLA violation and safeguard the QoS requirements.
- Placement of selected VMs in appropriate destination host for dynamic migration.

Our proposed VM consolidation technique includes four algorithms for aforementioned phases.

The System Model

In our work, the architecture is applied on IaaS environment that contains large scale cloud data centres which consist of thousands of heterogeneous Physical Machines (n) (PM = {PM_1, PM_2,…..PM_n}) . Each PM is characterized by R categories of resources for instance the CPU (multi core CPUs) performance (defined in Millions Instructions per Second (MIPS)), disk storage capacity, amount of RAM and network bandwidth. Numerous Virtual Machines (m) (VM={VM_1,VM_2,…..VM_m}) can be consigned to each PM with the help Virtual Machine Monitor (VMM) module which is otherwise called hypervisor. The system storage is designed as Network Attached Storage (NAS) . The NAS is organized and managed with a browser utility program which is widespread in clouds for the reason that NAS facilitates the allocation of live VM migration. The structure is completely illiterate as regard as the workload of PM and time of provisioning of the VMs. The application requirements for above mentioned constraints of PM are to be submitted towards the structure by individual clients of dispersed geological regions, and their demands are endowed with one or more different VMs.

Cloud functions have a wide category of workload types, ranging from High Performance Computing (HPC) to web-applications. The Cloud Service Providers (CSP) formulates an SLA deal with clients based on QoS requirements, and they must forfeit with a fine if there is any violation of SLA. The software layer contains three modules – Local and Global managers and dbmodule(to store the configuration data of all the VMs given by different Local Mangers of the whole hosts of the data centre). Consumers or behalf their brokers send requests to the Global manager - residing in the master node - in charge of negotiating the new requirements and gathers the facts from local managers to execute the resource utilization as well as issuing instructions for the VM placement optimization. While, local manager resides as a unit of the Virtual Machine Monitor (VMM) on each host of the IaaS architecture. The local manger is responsible concurrent scrutinizing and organizing all the host resources principally CPU performance. Indeed, the VM consolidation algorithms are employed in this module. The local manager scrutinizes the host resources and reaches a conclusion about necessity of VM migration and allotment based on the available resources with the help of VMM. Virtual machine manager decides which PMs are to be started, changed to be sleep-mode, or turn off.

Power model

Power consumption by a server in data center is connected with its processor, RAM, hard disk, and bandwidth. Latest studies [10] have exemplified that even if the DVFS policy is applied, the energy utilization by servers has a linear relationship with total electricity consumption along with CPU utilization. The energy consumption by a server is growing upward with idle CPU utilization status to fully CPU utilization status. Consequently, the server electricity consumption is designed by a linear function of its present CPU utilization (u) as:

$$P(u) = (P_{idle} * P_{busy}) + (1 - P_{idle}) \times u \quad (1)$$

Where, P is the anticipated power consumption, P_{busy} and P_{idle} are the power consumption value once a server is at its maximum utilization state and idle state respectively.

Overview of Encore an (Energy Conscious Linear Regression Analysis) Algorithm

EnCoReAn is the collection of three different algorithms such as UPReAn (for Predicting the Utility of a host), Overload and Under-load detection algorithms, and ECGCD (*Energy Conscious Greeny Cloud Dynamic)* is the package of EnCoReAn, VM Selection and Allocation algorithms. EnCoReAn algorithm utilizes the linear regression analysis to approximate a prediction function based on the historical data of CPU utilization above one hour back.

1. Linear Regression Analysis

Regression analysis is a statistical technique for quantitative data analysis which is useful for evaluating multiple independent variables as input for obtaining multiple depended variables as outcome by predicting the forthcoming values of data. This method is exensively used for predictions in different fields. As a result, it is particularly useful for assess and adjusting for confounding. Regression is of two type simple regression by inputting one variable and multiple regression by inputting more than one variable. The goal of regression is to approximate a regression function either linear or non-linear. The linear regression crafts a association among input variable (independent variable) 'x' and output variable (dependent variable) 'Y' with the aid of a straight line. The simple linear regression equation is as follows:

$$E(Y|x) = \beta_0 + \beta_1 x + \varepsilon \quad (2)$$

Where E(), which is to be read as "expected value of"; $Y|x$, which is to be read as "Y given x", points out that we are watching at the probable values of Y when x is limited to any single value. In other words, Y is the expected or predicted value (depended variable) of the result; where x is the predictor (independed variable), $\beta_0$and β_1 (regression coefficients) are the projected Y - intercept, is the projected slope, respectively. The Y-intercept as well as slope are forecasted from the model data so as to decrease the sum of the squared differences among the real values and the projected values of the output, i.e., the prediction reduces:

$$\varepsilon_i = Y_i - \hat{Y}_i \quad (3)$$

ε_i is the **residual which is the** difference among detected (Y) and estimated values($\hat{Y}$) of the outcome in a data point such as 'i'. Theoretically, if the values of x supplied a ideal prediction of Y then the distinction between detected (Y) and estimated values ($\hat{Y}$) of Y , i.e., residual ought to be 0. The equation of 'goodness of fitting line' is as follows:

$$E(Y|x) = \beta_0 + \beta_1 x \quad (4)$$

Then it is necessary to find a formula in which all points to be placed on the form.

Thus, reduction of the residual is an aim for achieving regression coefficients.

The widely accepted method to lessen the residual is the least squares method, in which the coefficient parameter of the model is selected such that the sum of the squared residuals whole data points is lessened. In simple term, Least Squares methods means the forecasts of the Y-intercept and slope decrease the sum of the squared residuals**.** It minimizes:

$$S_{re} = \sum_{i=1}^{m} \varepsilon_i^2 = \sum (\hat{Y}_i - \beta_0 + \beta_i x_i)^2 \quad (5)$$

Where, $S_{re}$is called the sum of the square of the residuals. We exactly to find out the values β_0 and β_1 that generate the sum of the squared prediction errors the least it can be.

$$\frac{\partial S_{re}}{\partial \beta_0} = 2 \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_i x_i (-1) = 0 \quad (6)$$

$$\frac{\partial S_{re}}{\partial \beta_1} = 2\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_i x_i(-x) = 0 \quad (7)$$

$$\beta_0 = \frac{1}{n}\sum_{i=1}^{n} Y_i = \frac{\widehat{\beta_1}\sum_{i=1}^{n} x_i}{n} = \hat{Y} - \widehat{\beta_1}\hat{x_i} \quad (8)$$

$$\beta_1 = \frac{\sum_{i=1}^{n} x_i Y_i = \frac{1}{n}(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} Y_i)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2} = \frac{\sum_{i=1}^{n}(x_i - \hat{x})}{\sum_{i=1}^{n}(x_i - x)^2} \quad (9)$$

Where, $\hat{x}$ and $\hat{Y}$ are the means of x and Y observations, $\widehat{\beta_0}$and $\widehat{\beta_1}$ are predictions of β_0 and β_1 correspondingly.

4. Methodology

Migration is the practice to move the Virtual Machine from one host to another host when a host is either overloaded or under-loaded. Migration processes contain three paces. -

1. When to trigger VM migration?
2. Which and what number of VM(s) to migrate?
3. Which host should be selected the VM to be placed?

When to Trigger VM Migration?

Migration is to be implemented when a host is either under-loaded or overloaded. A threshold value is used to identify the above situation. In this paper we are finding three threshold values i.e. upper threshold, prone-to-threshold and lower threshold which is used to make a decision about the time of migration. When the load of the host is more than the upper threshold, system is considered as overloaded, when the load of the host is in between upper threshold and normal state, the system is considered as prone-to-threshold and when the load on the host is less than the lower threshold , system is considered as under loaded. In all the cases migration is essential to stabilize the workload of the system as well as guaranteeing energy efficiency. The two steps before a feasible migration are:

A. Utility Prediction of the host.
B. Threshold calculation to find the under-loaded or overloaded host.

Utility Prediction of the Host

For the load calculation, we examine three parameters i.e. CPU, memory and bandwidth for the load calculation. Each VM has its own CPU, memory and bandwidth. Load on the VM can be calculated as:

$$\text{vm(cpu)}_{\text{usage}} = \frac{\sum \text{VM}_j^{\text{mips}}}{\sum \text{PM}_i^{\text{mips}}} \quad (10)$$

$$\text{vm(bw)}_{\text{usage}} = \frac{\sum \text{VM}_j^{\text{bps}}}{\sum \text{PM}_i^{\text{bps}}} \quad (11)$$

$$\text{vm(ram)}_{\text{usage}} = \frac{\sum \text{VM}_j^{\text{ram}}}{\sum \text{PM}_i^{\text{ram}}} \quad (12)$$

$$vm_{Util} = vm(cpu)_{Usage} + vm(bw)_{Usage} + vm(ram)_{Usage} \quad (13)$$

The prediction of the future resource requirements is crucial for effective resource execution in CDCs. The energy utilization by servers has a linear relationship with total electricity consumption along with CPU utilization, memory and bandwidth. In this work, we are calculating energy consumption of servers based on the afore mentioned multiple parameters. But Utilization of CPU plays a vital role in the VM load. So VM load can be directly proportional to the CPU usage of the VM.

$$VM_{Load} = VL = \frac{\sum VM_j^{mips}}{\sum PM_i^{mips}} \quad (14)$$

Total load on the host machine is calculated as the total load of the VM running into that host. If there are m VM on n^{th} host, then average load on the nth host.

$$PM_{Load} = \frac{\sum_{j=1}^{m} VL_j}{m} \quad (15)$$

We will not get the accurate host utilization by summing up the CPU, memory and bandwidth directly. Though we are calculating the host utilization based on the multiple factors such as CPU, memory and bandwidth by taking the product to combine them from virtual and physical machines. The host utilization can be calculated by the given equation:

$$h(u) = \frac{\boldsymbol{vm_j(cpu)_{usage}}}{\boldsymbol{1-pm(cpu)}} * \frac{\boldsymbol{vm_j(ram)_{usage}}}{\boldsymbol{1-pm(ram)}} * \frac{\boldsymbol{vm_j(bw)_{usage}}}{\boldsymbol{1-pm(bw)}}, j = 1,2,\dots,m \quad (16)$$

The proposed algorithm - EnCoReAn, at first, predict the CPU utilization of the host of last 24 hours with 5 minutes interval. Then it detects whether the host is overloaded, chance to overload, or under loaded to perform migration based on the three parameters such as, ram, bandwidth and CPU of the host. In both cases, the proposed method uses linear regression algorithm. The algorithm estimates a prediction function in accordance with the help of the linear regression method. The function demonstrated the linear relationship among the upcoming and present CPU utilization of each and every hosts are as following:

$$Y = \beta_0 + \beta_1 x \quad 17)$$

Where, β_0 and β_1 are regression coefficient constraints that predict in accordance with the *d* previous CPU utilization in a host. The *Y* is the predicted or expected value (depended variable) of the CPU utilization; x is the predictor (independed variable) which gives current CPU utilization value. The parameter d is assigned as 12 in our model, as the time gap of utilization dimensions is 5 minutes. The pseudo code of CPU Utility Prediction is follows:

Uprean (Utility Prediction Linear Regression Analysis) Algorithm

Input: hostList, Output: utilPredict
//Approximation of the prediction function as regard as 'n' former utilization
Initializing the β_0 and $\beta_1$with a random small value
for i=0 to n-1 do
for k=i+1 to n do
Get value of x_i from hostUtilHistory(i);
//equation(16)
Get value of y_k from hostUtilHistory(k);
//equation (16)
$\widehat{y_k} = \beta_0 + \beta_1 * x_i$
$\varepsilon = \sum(y_k - y_k)^2$ *// compute the lost function*
Calculate β_0 and β_1 by (8) and (9) respectively
Repeat steps 3 to 9 until the value of i>n
end for
end for
$utilPredict = \beta_0 + \beta_1 * currentTotalUtil$ *//Use the regression function: equation (2)*
return utilPredict;

Calculate the Upper Middle and Lower Threshold to Find the Overloaded, Prone-to-Overloaded and Under Loaded States

Three threshold values - such as lower, middle (prone-to-upper) and upper are used to describe the overloaded, prone-to-overloaded and under-loaded host. These values can be Static or Dynamic. In the case of static threshold these thresholds are

predestined and they will not be modified during the runtime, while in the case of dynamic threshold, these threshold values will be altered during the execution time.

Hence, dynamic threshold is more appropriate for the cloud, where resources demanded by the VM are changed dynamically and the workload submitted by the users of dispersed geographical area cannot be predictable.

VM migration prospects are raised with these dynamic threshold values. More precisely, the higher the upper threshold and lesser the lower threshold ought to be increased the possibility of the migration.

In the majority of the work, consider CPU usage is only aspect which is used to calculate the load, but RAM and bandwidth are the most critical elements in the system as compared with the CPU.

Hence, for calculating the upper threshold, in this paper we consider the three parameters such as CPU, RAM and bandwidth with equal intensity.

Upper Threshold Calculation

$$T_c = \frac{\sum_{j=1}^{m} VM_j^{mips}}{\sum PM_i^{mips}} \quad (18)$$

$$T_r = \frac{\sum_{j=1}^{m} VM_j^{ram}}{\sum PM_i^{ram}} \quad (19)$$

$$T_b = \frac{\sum_{j=1}^{m} VM_j^{bps}}{\sum PM_i^{bps}} \quad (20)$$

$$T_t = \sum(T_c , T_r , T_b)/3 \quad (21)$$

Overload detection algorithm

Input: hostList
Output:overld_hostList,cndt_hostList
d=12
get e from UPReAn(hostList)
get the value of T_t from equation(21)
T_u=(1-e T_t)//upper threshold calculation*
for host ∈ hostList do
for i=1 to d-1do
get currentTotalUtil from UPReAn(hostList)
get Utilpredict from UPReAn(hostList)
if((currentTotalUtil >= T_u or (Utilpredict > currentTotalUtil))
1overld_hostList ← host //host becomes the status of overloaded
end if
T_c=1-eT_u*
if (currentTotalUtil>= T_c)
cndt_hostList← host //host is under pressure to be overloaded very soon
Host is not able to assign any new VM
end if
return overld_hostList
return cndt_hostList

After gathering the records of 12 last host utilization data, EnCoReAn algorithm resolves the expected host usage of future. We presume a host is overloaded either the current utilization is greater than upper threshold value(T_u) (calculated by taking the average of three parameters such as CPU, memory and bandwidth utilization(T_t) of all VMs of single host *(Equation (14) to (17))* by taking a percentage of the T_t with the help of intensity parameter which is the '*e*' which is the percentage of the host load *(Equation(15))* of the total utilization or the prediction utilization value is above the available total capacity of the host (by considering multiple factors).

Then using the algorithm, it is find out the host which are near to an overloaded condition (candidate as an overloaded host), by taking a percentage of the upper threshold value (T_u) with the help of intensity parameter '*e*' which is the percentage of the host load (Equation(15)).

In both cases '*e*' acts as a safety parameter which maintains the tradeoff between the number of VM migration and resources wastages. Lower the value of '*e*' will result the lowering of the power consumption but higher will create the SLA violations and vice-versa. On the basis of the previous studies '*e*' is 5% ie.0.05.Threshold for the next interval is analyzed that is derived from the historical data of host i.e. threshold. The T_r interval highly proportional to the PM utilization in the T_c interval. This host is exclusive of assignment of new VMs. These measures will help to avoid SLA violation by the movement from and placement in unhealthy PMs of the VMs. We employ UPReAn in the under-loading detection strategy to determine an under-loaded host.

Under Load Detection Algorithm

Input: hostList
Output:undrld_hostList
k=12
for host ∈ hostList do
get currentTotalUtil from UPReAn(hostList)
get Utilpredict from (UPReAn(hostList)
*T_l =0.3*totalUtil(host)*
for i=1 to k-1do
if(currTotalUtil(host)>=0 or Utilpredict<= T_l)
undrld_hostList ← host
end if
end for
end for
return undrld_hostList

After gathering the records of 12 last CPU utilization data, EnCoReAn algorithm resolves the expected CPU usage in a host. If the CPU utilization of a host equivalent of either zero or when the prediction utilization will be less than or equal to 30% of total host utilization, the host is assumed as under-loaded host. So as to decrease the energy consumption of that particular host it essential to migrate all VMs to another hosts. Then the host toggles to the sleep- mode after placing all VMs. The destination host should satisfy three condition while accepting the VMs from either overloaded or under-loaded host:

1. Not an overloaded host.
2. Not a candidate for being overloaded.
3. Not an under-loaded host.

Which and What Number of VM(S) to Migrate?

It is very crucial task to decide which VM should be migrated because it is the main factor of causing total migration time and down time. Down time means the time for which VM is not accessible to the user, where total migration time means time entailed to shift the VM to another host completely. If the huge VM is selected, as a consequence the total migration time as well as down time will be increased. If we select smallest machine, then a large number of VM should be migrated. So in our approach, we arrange the VM in descending order in which the size of VMs are greater than or equal to the difference among the entire host utilization and upper threshold[18]. Besides that we are using a new approach to select the VM to be migrated in descending arrangement of their predicted response time. Response time is calculating by using the historical data provided by the local manager to detect the old VM which is the identical to the new VM by decomposing the components.

Virtual Machine Components Similarity Checking Strategy

There are *m* old VMs. Suppose, if we insert a new VM, the configuration of the VM data ought to be passed by different local

managers of different hosts in the data centre will be placed into the dbmodule (database module). We assume there are many components such as CPU, memory, disk space, I/O rate, bandwidth etc. of each VM whether it is new or old. Here, we are taking 3 components out of several such as CPU, memory and bandwidth. The group of *m* former virtual machines is denoted as VM= {VM_1, VM_2,…..,VM_m}. We are using Euclidean distance formula for calculating similarity of new VMs' components as compared with new VMs components. For finding the similarity between two same components in the old and new VM we are using Euclidean Distance formula The lower the distance between 2 points such as *oc* (old VMs' component) and *nc* (new VMs' component), then the higher the similarity.

Algorithm for Response Time Calculation

Function Restime ()
Input: Old VM's (VM_i with its components{ oc_{i1},oc_{i2},….,oc_{in}} and its response time such as{ t_{i1}(c_{i1}),t_{i2}(c_{i2}),…., t_{in}(c_{in})} i=1,2,3,…n
Output: Response time of components of new VM
Decompose the new VM into several components such as VM_i{nc_{i1},nc_{i2},…nc_{im}} j=1,2,3,…m
Use historical data to input the old VM (VM_i with its components{ oc_{i1},oc_{i2},….,oc_{in}} i=1,2,3,…n and its response time such as{ t_{i1}(c_{i1}),t_{i2}(c_{i2}),…., t_{in}(c_{in})}
for each nc(VM_i) in oc(VM_i), do
$d(nc_i - oc_i) = \sqrt{\sum_{i=1}^{m}(nc_i - oc_i)^2}$*//Euclidean distance formula for calculating similarity*
Find the VM with same configuration
Calculate the Responsetime(VM_i{nc_{i1},nc_{i2},…nc_{im}})
Return Rtime //predicted response time of new VM
End

The PSEUDO CODE for the VM Selection Procedure

*Function **Energy Conscious VM Placement ()***
Input: hostList, vmList
Output:migrationMap
4.Restime(Rtime)=max
for host ∈ hostList do
get hostUtil from host.util()
if (hostUtil ≥ T_c) then
while hostUtil<T_c
for each vm in vmList do
get vmUtil from vm.util()
x=hostUtil-T_u
migratableVms = getMigratableVms (host)
optimalVms = vmList.srtByDesc (migratableVms,key:vmUtil-x)
optVmList.add(optimalVms)
optimalVms= optVmList.srtByDesc(optimalVms, key:Rtime)
vmsToMigrate.add(optimalVms)
host.remove(optimalVms)
end for
hostUtil =hostUtil − optimalVms.util()
end while
end if
if (hostUtil < T_l) then
for vm ∈ vmList do
vmsToMigrate.add(host.getVmList( ))
vmList.remove(host.getVmList())
host.remove(VmList)
host.sleep()
end for
end if
end for
migrationMap.append (getNewVmPlacement (vmsToMigrate))
vmsToMigrate.clear( )
return migrationMap
End function

Which host Should be Selected the VM to be Placed?

In this section we demonstrate an algorithm for VM placement. VM placement is the current subject matter for many research works. When a VM is to be opted for migration, it is essential to find out an appropriate host based on VM characteristics and overall policy pursued in the data center.
Following the VM selection, subsequently select the host where the selected VM will be placed. The VM Placement problem can be scrutinized as a bin-packing problem where bins of varying sizes representing the physical machines and the VMs are considered as the items to be placed in the bins. [18].Bins sizes are the existing CPU capacities of the PM and charges are consequent to the electrical energy consumption by these PMs. It is a NP-hard decision problem in nature. Erroneous choice of the PM may raise the number of VM migration which cause resource wastage and SLA violation. In our work we select the VM which is energy conscious based on the optimum threshold value of the host as well as least power consumption.

Algorithm for VM Allocation to A Host

Input: hostList, vmList Output: VMsallocation
vmList.sortDecendingCPUutil()
for vm ∈ vmList do
leastpow=MAX
excluded_Host= overld_Host+cndt_Host+slp_Host
assigned_Host=host-excluded_host
Pow=power(host)
for host ∈ hostList do
if host.util() <=T_C && host.util()>=T_l then
while host has free resources for vm then
pow=pow+ predictedPower(host,vm)
if(pow<leastpower) then
assigned_Host = host
leastpow=pow
if(assigned_Host = NULL) then
assigned_Host= Assign(slp_Host)
repeat step 7 until assigned_Host!=NULL
VMsAllocation.add(vm,assigned_Host)
else then
VMsAllocation.add(vm,assigned_Host)
return VMsallocation

5. Performance appraisal

Workload Data

To facilitate to evaluate the ECGCD Algorithm, the workload data, which are used for simulation, that are extracted from the CoMon Project, a widely scalable monitoring scheme for PlanetLab in CloudSim [19]. We employ the similar power models supplied in the website for both servers as shown in the following Table 1.

Table 1: The Power Consumption at Diverse Load Levels in Watts

Server	Load										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProliantG4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProliantG5	93.7	97	101	105	110	116	121	125	129	133	135

The features of PMs and VMs used in the researches are listing in Table 2 and Table 3, respectively.

Table 2: Host Parameters

Parameters	Server	
	HP ProliantG4	HP ProliantG5
Number of Host	400	400
Number of Cores	2	2
MIPS	1860	2660
RAM	4096	4096
BW	1GB	1GB
Storage	1.5GB	2GB

Table 3: VM Parameters

Parameters	VM Type			
	High-CPU Medium Instance	Extra Large Instance	Small Instance	Micro Instance
Number of Cores	1	1	1	1
MIPS	2500	2000	1000	500
RAM	870	1740	1740	613
BW	1MB	1MB	1MB	1MB
Storage	3.85GB	2GB	1.75GB	613MB

The real workload traces provides by CoMon Project for CPU utilization data of more than a 1000 virtual machines from over 500 physical machines, with the interval of 5 minutes for CPU utilization for the measurement, from the servers that are situated all over the world. We have randomly selected each VM's workload traces from any one of the VMs of 3 days and numbers of VMs that are surveyed on each day are shown in following Table 4.

Table 4: The Number of VMs in the Real Workload

Date	Number of VMs
3rd March	1052
25th March	1078
12th April	1054

Experimental Setup

We have proposed a large-scale IaaS environment which provides tremendous computing resources for its users. For the evaluation the effectiveness of our experiment, executions have been performing on the CloudSim simulation toolkit. In the cloud computing community, CloudSim is becoming progressively more popular due to their assistance for elastic, scalable, proficient and repeated evaluation of resource provisioning procedures for various applications [20]. The proposed algorithm ECGCD is evaluated with three standard over load detection algorithms such as MAD (the median of the absolute values of the differences among the data values and the overall median of the data set) and IQR (which determines the threshold of a host to be spotted as overloaded by using inter quartile range) both are using dynamic threshold values and LR (depends on the LOESS local regression algorithm, aims to settle on the upper threshold by observing a regression curve that estimates the CPU utilization) with two different VM selection algorithms – Minimum Migration Time (MMT) and Random Choice (RC).

We have been simulated a wide-ranging data center that consists of 800 heterogeneous physical servers, on the half of which are HP Proliant ML 110G4 servers, and the other half comprises of HP Proliant ML 110 G5 servers. The power consumption features of the servers are displayed in Table 1. The characteristics of PMs and VMs are depicted in Table 2 and 3 respectively. In the beginning, VMs are equipped based on their resource demands and over-utilization of VMs are allowed. Meanwhile of the simulation, VMs rearranges their resources requirements and create dynamic consolidation chances.

Table 5: Simulation results of benchmark algorithms and ECGCD (mean values)

Algorithm	Energy (kWh)	Number of VM Migration
MAD-MMT	195.316	26835
MAD-RC	197.231	27964
IQR-MMT	201.106	28876
IQR-RC	204.215	28340
LR-MMT	172.308	31758
LR-RC	174.157	30232
ECGCD	117.415	13109

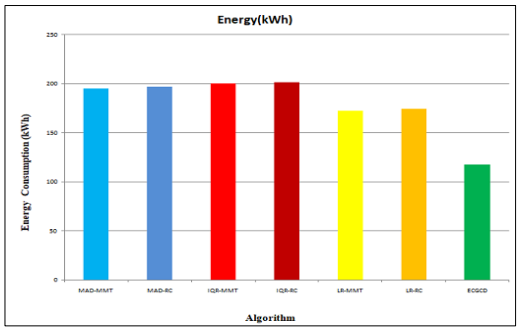

Figure 1: Energy consumption by ECGCD and other benchmark scheme for random workload traces (mean values)

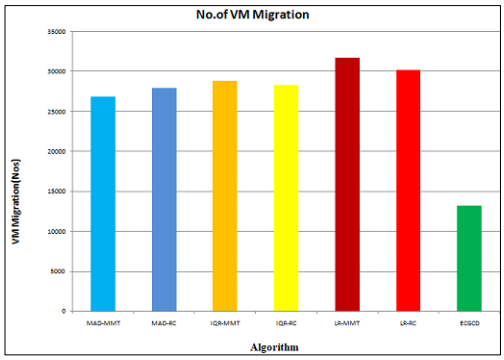

Figure 2: VM migration by ECGCD and other benchmark scheme for random workload traces (mean values)

Our propsed algorithm ECGCD showing least energy consumption as compared with three benchmark algorithms with two vm selection techniques as shown in Figure 1. The algorithm ECGCD, accessing energy saving around 40%,41%,42%,43%,32%,33% as weighed against MAD-MMT, MAD-RC, IQR-MMT, IQR-RC, LR-MMT, LR-RC as the mean value of three random date values respectively. Moreover, the number of migration is also reduced in considerable amount while comparing with the aforementioned benchmark algorithms (Figure 2). This achievement can be obtained due to the findings of three threshold values such as lower, middle and upper thresholds for changing unused servers to low-power mode (i.e., sleep or hibernation) to attain the least power consumption.

6. Conclusion

Dynamic consolidation of Virtual Machines is an effectual means to advance the resources utilization as well as energy efficiency in Cloud Data Centres. This technique plays an important role of reducing energy consumption which might bring down the CO_2 emission and boost up ROI for cloud service contributors by means of turning off the idle host machines. In this paper we have presented Energy Conscious Dynamic VM Consolidation with auto adjustment of three threshold values such as upper threshold, middle (prone-to-upper) and lower threshold. We have assessed the proposed algorithm for a large-scale IaaS environment during substantial simulation on CloudSim 3.0 toolkit. PlanetLab workload of CloudSim toolkit is used in the simulation. It has been selected as a simulation platform because as it provides all the services of Cloud computing architecture. We have customized the simulator by itself to achieve our research work in Cloud platform. To check the efficiency of our approach we compare ECGCD, the proposed one with three standard overload

detection algorithm-MAD, IQR and LR- with two section algorithm - MMT and RC presented in Cloudsim toolkit. The proposed approach proves the improvement of the data center resources utilization and diminishes energy consumption.

As a future enrichment, we can propose a new innovation, which diminishes the trade-off among power utilization and better QoS performance. We will also be introduced the process of implementing the new holistic approach for VM consolidations to reduce the migration cost and scalability of this approach by considering the network resource utilization and traffic to optimize VM placement in an optimal host.

References

[1] Jansen W & Grance T, "Guidelines on Security and Privacy in Public Cloud Computing", *National Institute of Standards and Technology*, (2011), pp.800-144.

[2] Garcıa PA, Fernández JMM, Rodrigo JLA & Buyya R, "Proactive Power and Thermal Aware Optimizations for Energy-Efficient Cloud Computing", *Escuela Tecnica Superior De Ingenieros De Telecomunicacion*, (2017).

[3] EC-European Commission. (2007). *Limiting Global Climate Change to 2 degrees Celsius. The way ahead for 2020 and beyond*. COM/2007/2.

[4] Murugesan S & Gangadharan GR, *Harnessing Green IT: Principles and Practices*, Wiley Publishing, (2012).

[5] Buyya R, Broberg J & Goscinski AM, *Cloud Computing: Principles and Paradigms*, John Wiley & Sons, (2010).

[6] Abali B, Canturk I, Jeffrey OK, Suzanne KM & Dipankar S, *Live Virtual Machine Migration Quality of Service*, U.S. Patent Application, Vol.15, (2017).

[7] Abdullah M, Lu K, Wieder P & Yahyapour R, "A Heuristic-Based Approach for Dynamic VMs Consolidation in Cloud Data Centers", *Arabian Journal for Science and Engineering*, (2017), pp.1-15.

[8] Quanwang W, Fuyuki I, Qingsheng Z & Yunni X, "Energy and Migration Cost-Aware Dynamic Virtual Machine Consolidation in Heterogeneous Cloud Datacenters", *IEEE Transactions on Services Computing*, (2016).

[9] Georgia K, Emmanouil S, Amir AS & Polychronis K, "Can Everybody be Happy in the Cloud? Delay, Profit and Energy-Efficient Scheduling for Cloud Services", *Journal of Parallel and Distributed Computing*, Vol.96, (2016), pp.202-217.

[10] Mehiar D, Bechir H, Mohsen G & Ammar R, "An Energy-Efficient VM Prediction and Migration Framework for Overcommitted Clouds", *IEEE Transactions on Cloud Computing*, Vol.7, No.4, (2017).

[11] Zhe H & Danny HKT, "M-Convex VM Consolidation: Towards a Better VM Workload Consolidation", *IEEE Transactions on Cloud Computing*, Vol.4, No.4, (2016).

[12] Fahimeh F, Tapio P, Pasi L, Juha P, Nguyen TH & Hannu T, "Energy-aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model", *IEEE Transactions on Cloud Computing*, (2016).

[13] Michael P, Gavriil T & Alex, D, "Decentralized and Energy-Efficient Workload Management in Enterprise Clouds", *IEEE Transactions On Cloud Computing*, Vol.4, No.2, (2016).

[14] Mohammad AK, Mohd ND, Azizol A, Shamala S & Mohamed O, "Energy-Efficient Algorithms for Dynamic Virtual Machine Consolidation in Cloud Data Centers", *IEEE Transactions on Green Cloud and Fog Computing: Energy Efficient and Sustainable Infrastructures, Protocols and Applications*, Vol.5, No.69, (2017), pp.10709–10722.

[15] Jungmin S, Amir VD, Rodrigo NC & Rajkumar B, "SLA-aware and Energy-Efficient Dynamic Overbooking in SDN-Based Cloud Data Centers", *IEEE Transactions on Sustainable Computing*, Vol.2, No.2, (2017).

[16] Yang Y, Xiaolin C, Jiqiang L & Lin L, "Towards Robust Green Virtual Cloud Data Center Provisioning", *IEEE Transactions on Cloud Computing*, (2017).

[17] Nguyen TH, Di FM & Yla-Jaaski A, "Virtual Machine Consolidation with Multiple Usage Prediction for Energy-Efficient Cloud Data Centers", *IEEE Transactions on Services Computing*, (2017).

[18] Yi H, Jeffrey C, Tansu A & Christopher L, "Using Virtual Machine Allocation Policies to Defend against Co-resident Attacks in Cloud Computing", *IEEE Transactions On Dependable and Secure Computing*, (2015).

[19] Dan K, Sasa M, Matej G & Ondrej P, "Testing Internet Applications and Services Using PlanetLab", *Computer Standards & Interfaces*, Vol.53, (2017), pp.33-38.

[20] Rodrigo NC, Rajiv R, Anton B, Cesar AF, De R & Rajkumar B, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", *Software: Practice and Experience*, Vol.41, No.1, (2015), pp.23– 50.