International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-2S, December 2018

Enbase: Energy- Conscious SLA-Oriented Dynamic VM Consolidation Heuristics in Green Data Centres Using Ant Colony System

Sajitha.A.V , A.C.Subhajini

Abstract: ***Cloud computing is an burgeoning technology which offers scalable and on-demand services to huge set of users in variety of domains globally. As the popularity increases, the complication of the cloud environment is also increasing at a massive scale. The foremost challenge of this technology is high amount of energy consumption due to the load of servers. Moreover this crisis, it will not provide fruitful results also. In general, each consumer has a service level agreement (SLA), which states some constraints over performance and/or quality of service that it obtains from the method. The breach of the affirmation may lead to SLA violation between consumer and provider. Dynamic Virtual Machine consolidation (DVMP) offers a momentous prospect to conserve energy in the data centers . A VM consolidation technique employs live migration of Virtual Machines(VMs) in order that the underloaded Physical Machines (PMs) can be turned-off or set into a least-power mode and overloaded PMs are reducing its load. But the VM consolidation in live migration may cause the increase of total migration time as well as down time which again grounds the breach of SLA. In this scenario, we proposed a multi-objective SLA oriented energy efficient and network aware DVMP algorithm which utilizes Ant Colony Optimization meta-heuristic named enBASE which finds out global best migration plan to ensure the migration to increase the energy efficiency, minimization of SLA violation as well as improvement of the scalability of the system. The simulation results prove that the proposed algorithm presents an efficient as well as effective solutions for Dynamic VM consolidation inside the cloud data centers. Furthermore, this outpaces three benchmark algorithms such as two ant colony optimization based and one BFD based VM consolidation algorithms in respect of increase in energy efficiency, and reduction in total migration time, down time while preserving SLA violation minimization.***

Keywords: Cloud Computing, Dynamic Virtual Machine Placement, Live Migration, Total Migration Time, Down Time, Ant Colony System, Service Level Agreement

I. INTRODUCTION

Cloud computing is an blooming across-the-board distributed computing technology for providing elastic resources in the class of computing services such as software, databases servers, storage, networking, analytics etc.

Manuscript published on 30 December 2018.
***** Correspondence Author (s)
Sajitha. A. V, Research Scholar, Department of Computer Applications, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India email: sajithaav09@gmail.com
Dr. A. C. Subhajini, Assistant Professor, Department of Computer Applications, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India email: acsubhajini@yahoo.com

on demand to its enormous number of clients from geographically scattered area with 24X7 in 365 days via common internet protocols.

Companies contributing these services are called cloud service providers and generally they are liable for cloud computing services rely on usage, comparable to the other utility service payment like water, telephone or electricity at home on metered basis[1]. Cloud provides Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) as service models. It also consists of four deployment models such as private, public, community and hybrid clouds. While the popularity of the cloud computing in every fields increases, whether it is a multinational company or sole proprietary ship or whether it is public or private organization or whether it is official or personal affairs, the various cloud service providers for instance Google, Amazon, Microsoft, Yahoo, Apple etc. are persuading to deploy large number of energy hungry data centres in dispered geographical area world wide. Each data centre is a centralized repository consists of thousands of physical machines organized in hundreds of racks that can run millions of Virtual Machines. It could consume as much power as a small hydroelectric power station could produce. The environmental impact is the CO_2 emission, which makes up greatest share of the Green house Gas (GHG) into the atmosphere. According to the report of European Union, prior to the year 2020 to maintain the global temperature raise, lower than 2 °C, a decline in CO_2 emission volume of 15%-30% is needed [2].

Thus, energy efficiency is a great challenge in the management of data centres now-a-days. Virtualization is a key technology of cloud computing which acts as a powerful tool for eliminating energy inefficiency by employing multiple Virtual Machines (VM) in a single Physical Machine(PM) via live VM migration practices. VM consolidation is a practice of minimizing the number of running PMs by migrating and consolidating the VMs into the reduced number of PMs[3]. VM consolidation includes VM placement (the method of selecting the most appropriate host for the given VM) and VM migration (is the task of transferring a VM from one PM to another).

To minimize the energy consumption, it is a good way to reduce the number of PMs. However, this will increase the load of PMs and can cause a system overload

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

which may be the reason for performance degradation. Another cause for the performance degradation is owing to live VM migration through the increase of total migration time and service downtime. This may be the grounds for the rejection of destination PM ie., destination is not be able access the VM, due to system fault or route failure. Anyhow, it will create topsy-turvy relationship in between the client and service provider. Before starting cloud services to the client, the cloud service provider will make an agreement with the client that ensures a minimum level of service is maintained. It guarantees levels of reliability, availability and awareness to systems and applications, while indicating who will manage when there is a service interruption[4]. This agreement is known as Service Level Agreement(SLA). Due to the unexpected service interruptions or failures, there is a chance of SLA violation. Thus to reduce the SLA violation, it should be compulsory to monitor and manage the VM migration while ensuring the migration is done in proper manner with in a period of time specified in the SLA. In this work, we tackle the Dynamic VM Consolidation Problem(DVMCP) with the goals of reducing energy utilization of data centres while minimizing the SLA violation and satisfying the QoS constraints. Here, we present a distributed cloud computing system architecture to execute DVMP by finding out the a better route for accessing optimal PMs to trim down their energy consumption as well as perform mandatory live VM migration. For that we have been proposing a multi objective DVMC approach that employs a extremely adaptive online optimization meta-heuristic algorithm entitled as Ant Colony System (ACS) to optimize the VM placement strategy. In our previous work In our previous work[5], we developed a bin-packing based model with linear regression strategy for dynamic VM consolidation algorithm called as Energy Conscious Greeny Cloud Dynamic (ECGCD) algorithm, is the mixture of several adaptive algorithms such as EnCoReAn (for Predicting the Utility of a host, Overload and Under-load detection, VM Selection and Allocation algorithms), which helps to attain live VM migration by switching-off unused servers to low-power mode (i.e., sleep or hibernation), as a result saves energy and efficient resource utilization. But the host is chosen only on the basis of their least energy consumption.

Major contributions of this paper:

1. We devise a multi-objective combinatorial optimization algorithm such as energy-efficient and network Balanced Algorithm for Service Level Agreement Ensuring -enBASE, by utilizing the Ant Colony System (ACS) metaheuristic for energy-efficient Dynamic VM Consolidation to optimize 4 distinct objectives such as improving of energy efficiency, minimization of total migration time as well as downtime and improving SLA by avoiding SLA violations.
2. While using this algorithm, finding out the shortest route with congestion free minimum bandwidth path (in connection with page dirty rate in pre-copy live vm migration) for a target host with least power consumption, highest bandwidth, sufficient memory and storage capacity.
3. A novel ACS based algorithm - Multi- OBjective Ant Colony System algorithm named MOBACoS -as a part of enBASE- suggests for finding best migration plan, which will reduce the computation time, and increase the scalability of the system due to the heuristic nature of this algorithm.
4. By applying enBASE, the pre copy live migration is monitoring continuously and ensuring the mandatory migration. As a consequence, it is reducing the migration time as well as downtime and assuring proper SLA execution. We are taking into consideration of multi-dimensional resource type consumptions of PMs for dynamic VM consolidation problem such as CPU, memory, storage along with network bandwidth.
5. The execution of the proposed enBASE approach is assessed by CloudSim simulation based on real workload traces with multiple factors and several performance metrics. We evaluated our planned strategy with the prevailing dynamic VM consolidation approach such as BFD in [29] and with the ACS-oriented VM consolidation scheme in [7,8]. The results exhibit that enBASE significantly optimizes the VM allocation and migration by outperforming the existing VM consolidation approaches in terms of energy consumption, total migration time , downtime, and percentage of SLA violations.

This paper is sorted out as in the following manner: Section II discusses some related concepts in connection with the SLA aware energy efficient dynamic VM consolidation problem with the support of the pertinent literature review. Section III narrates a finite outline of the system architecture. Section IV describes a multi- objective combinatorial optimization algorithm enBASE in conjunction with a Multi Objective ACS (MOBACoS) for finding global best migration plan for mandatory vm migration to ensuring energy efficiency, reducing migration time as well as down time and reducing SLA violation. Section V explains the performance evaluation and extensive scrutiny of the end results with other three existing algorithm with the help of real workload traces. Section VI reiterates the proposed work and discusses the future work directions.

II. LITERATURE REVIEW

At present, several efficient quotable researches are carrying out in the region of improving energy efficiency though proper utilization of resources in cloud data centres. It will helps to reduce the green house gas such as CO_2 emission and made an ecofriendly IT structure. Proper network route selection along with optimal PM selection for live VM migration will considerably trim down the total migration time along with downtime. In addition to that, the efficient usage of resources in cloud environment will noticeably reduce the Service Level Agreement violations. In this section, we review significant attempts suggested in the literature for attaining energy efficiency and improving SLA via proper resource utilization with the help of proper network routes in cloud data centers. Ferdaus, M. H. et al.[6], have been put forwarded an Ant Colony Optimization algorithm termed as ACO based, Migration overhead-aware Dynamic VM Consolidation (AMDVMC) as a solution to the dynamic VM consolidation dilemma with the aim to minimize data center resource wear and tear, energy consumption, and aggregate migration overhead.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

This algorithm is incorporated with the suggested decentralized consolidation scaffold and exploits the proposed migration overhead assessment sculpts.

Farahnakian, F et al**.** [7], have been formulated energy-efficient VM consolidation as a multi-objective combinatorial optimization dilemma rooted in Ant Colony System named as ACS-VMC to optimize three contradictory objects concurrently. The objects including the minimization of the energy consumption, reduction of the number of VM migrations, and avoidance of the SLA violations.

Liu, X. F. et al.[8], have been proposed an unified algorithm ground as an ant colony system (ACS), termed as the Unified ACS (UACS), selecting a sever regarding CPU capacity and energy consumption in both overloaded and underloaded situation for VM migration. With each of hosts, the UACS used to locate the possible solutions with the least number of VM migrations with the help of dynamic pheromone dropping strategy as well as a unique heuristic information approach .

Ashraf, A. et al.[9], have been discussed an algorithm for virtual machine (VM) consolidation in cloud based data centers based on multi-objective ant colony system with the goals of cutting down the number of working PMs and minimizing the number of VM migrations.

Aryania, A. et al.[10], have been proposed an Energy-aware VMC algorithm (EVMCACS)by applying Ant Colony System to resolve the problem as a multi-objective optimization to trim down the power consumption of data centers while increasing the number of inactive PMs and decreasing the amount of migrations by insisting on the quality of service necessities via SLAs.

Hassan, M. K et.al[11], have been proposed a machine learning technique with modified kernel along with Friedman rank summation and average ranking for VMs live migrations I accordance with an adaptive forecasting of utilization thresholds. It is ensuring the QoS and SLA in virtualized cloud based data centers for critical applications with strict SLA.

Mustafa, S et al.[12], have been presented two algorithms named Available Capacity and Power (ACP) and Required Capacity and Power (RCP) through which a server is opted for the base of obtainable CPU competence and the power that will be devoured for that capacity. While using the historical data and intelligently changing the values of the data based on varying workloads, it helps to achieve dynamic VM consolidation. They proved through the experimental result that it reduced the energy consumption as a consequence reducing the SLA violation due to the proper use of resources.

Su, W. et al.[13], have been evaluated Balanced SLA-aware tenant placement (BSP) algorithm in taking the benefit of the resource sharing. They have been built up a queuing network model to expose the tailored challenging of multi-tenant SaaS. As a result it reduced both over provisioning of resources and violation of SLA.

Anan, M. et al.[14], have been implemented an energy efficient computing framework- Dynamic Migration Algorithm(DMA)- for green cloud data centers that develops energy efficiency, decreases operational costs, and assembles essential Quality of Service. Through this algorithm, Virtual Machines in servers utilizes the data centres' servers privileged order. This automatically reducing the SLA violation via over deployment of resources.

Zhang, J et al.[15], have been formulated a multi-objective nonlinear programming taking into account the server cost and the network cost features concurrently in broad restraints, where VMs are either independent or correlated and both of them can be heterogeneous. Similarity based checking will helps to deploy the similar VMs to the remaining PM capacity which will reduced the fragment leak. This algorithm guarantees the QoS parameters while saving cost and obviously minimizes the SLA violation.

Jungmin, S. et al.[16], have been proposed Dynamic Overbooking algorithm which jointly 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 approximated resource allocation ratio derived from the past data scrutinizing from the online investigation of the physical host as well as network consumption with no prior-awareness about the workloads. They can be gained energy efficiency and a massive energy cost reduction by minimizing the misuse of excess allocated resources, then together reduced the SLA violation by distributing adequate resources for the real workload.

Nguyen T. H et al.[17], have been presented a VM Consolidation algorithm with multiple Usage Prediction (VMCUP-M) for enhancing power 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.

Mohammad,A.K.,et.al.[18], have been presented energy efficient and SLA-aware dynamic VM consolidation Mechanism (PCM) Algorithm which is the combination of four algorithms: Over loading host detection algorithm, Under loading host detection, VM Selection and VM Placement algorithms. In their approach they consider multiple parameters for forecasting 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.

Justafort, V. D.,et al.[19], have been explored an innovative mathematical representation, called CB_ OPT_ SLA, that cut downs the carbon footprint of an Inter-Cloud environment. This model guides to best possible configurations with least CO_2 emissions in an independent cloud, as well as in an Inter-Cloud environment, and producing a saving up to 65% that traditional model. Moreover, it proves that it yields the reduced rate of carbon foot print and prevent SLA violation while performing VM consolidation.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Fahimeh, F et al.[20], have been investigated Utilization Prediction aware VM Consolidation (UPVMC) - a Dynamic VM consolidation approach which creates a VM consolidation as a way of multi-objective vector packing dilemma. By the advent of a regression-oriented prediction form which is exercised for the prediction of current as well as prospect resource utilization by consolidating the VMs into the decreased number of active PMs. VM selection consider MMT (minimum migration time) strategy and VM allocation consider Modified Power Aware Best Fit Decreasing with SLA as an vital factor.

III. SYSTEM ARCHITECTURE

A. Data Center Model

FAT tree topology is the most accepted data center structural design which consists as three_tier trees of hosts, servers as well as switches[21]. This structural design (Figure 1) comprises of the core tier on the root of the tree, then the aggregation tier which is in charge of routing activity, and finally the access tier which contains the collection of computing hosts. In comparison with the former two_tier data centers which is not supporting above 5,000 hosts, the three_tier architecture can hold enormous computing servers, e.g., a Data Centre with the capability of 2,00,000 hosts.

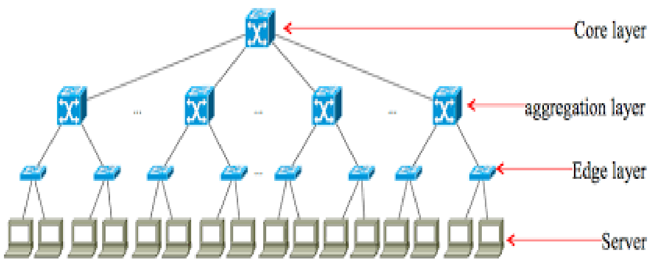

Figure 1: Fat-tree topology

Fat-Tree is suggested to apply commodity switches in data centers, which can strengthen any communication prototype with full bisection bandwidth. We characterize a data center by an undirected graph G=(V_G,E_G), where V_G is the set of vertices (i.e., PMs and switches) and E_G is the set of edges (i.e., links).For every VM, we are required to discover M_{vm} optimal hosts (PM_o) to sustain its physical resource necessities, and a set of path in G to direct its traffic between VMs. Each and every pathway contains of a set of links that inter-connect switches and PMs.

B. System Model

As specified in our previous work [30], the architecture is applied on IaaS environment that contains large scale cloud data centres which consist of thousands of heterogeneous hosts, the group of the hosts can be stated as $PM=\{pm_1, pm_2, ...pm_n\}$ which is represented in figure 2. The cloud data center is alienated into K number of regions, and all regions have total number of PM count cnt, so $n = K * cnt$. We define the network route among two pm s as the network hops among them, and shown it in equation 1,

$$l(pm_p, pm_q) = \begin{cases} 1, pm_p - pm_q; \\ l(pm_r, pm_q)+1, pm_p - pm_r. \end{cases} \quad (1)$$

The set of regions is defined as $O=\{o_1, o_2, ...o_i, ...o_K\}$.

The set of hosts in a region o_x is represented as $PM_x = \{pm_1^x, pm_2^x, ...pm_{count}^x\}$ and $VM_i^x = \{vm_1^i, vm_2^i, ...vm_m^i\}$ stands for the number of virtual machines in a host pm_i^x which belongs to o_x. Each host *i* is designated by R type of resources such as the CPU performance which is measured in Million Instructions Per Second (MIPS), the amount of memory, and the bandwidth (BW). These hosts contains multi- core CPUs. Virtual Machines can be dealt out to each host via Virtual Machine Monitor (VMM) otherwise called hypervisor. The Network Attached Storage(NAS) are configured and managed with a browser-based utility program which is applied to represent these host's local disks. These two characters of host serve the purpose of live migration. At first, VMs are set in a host arbitrarily. For that VMs are arranged with their resource usage indescending order. Because of the dynamic workloads, the VMs' resource utilizations may fluctuate periodically. In this manner, a basic proficient allocation approach should be expanded with a VMC algorithm that can be executed sporadically. Figure 2 portrays the proposed framework show that comprises of two categories of agents, such as Local and Global Manager [20]. A Local Manager (LM) stays as a component of the Virtual Machine Monitor (VMM) in a host to resolve the host condition recognition sub-problem by scrutinizing the present resource employments of the host. VM Placement is done by continuous monitoring of the PMs by the Local Manager of each PM. The PM is categorized as Normal(〖Nrml〗_host), Overload(Ovrld_host), prone to be overloaded (Cndt_host),and Underloaded(Undrld_host) based on its CPU, bandwidth and memory utilization.

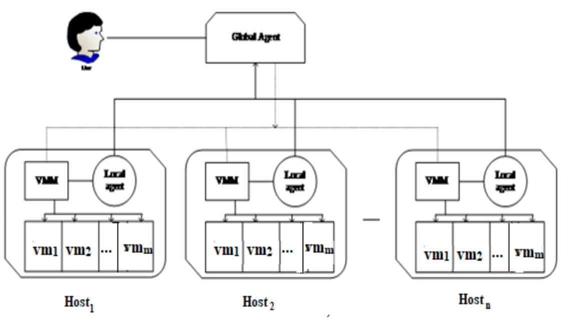

Figure 2: System model

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

These host categorization is done in our previous algorithm ECGCD[5].The Global Manager (GM performs as a manager and then optimizes the VM placement problem by executing enBASE - the proposed algorithm. GM collects all the details from each LM about the status of PMs which reside under the control of them. This data is considered to find the global best migration plan by means of MOBACoS.

C. Power Model

Power consumption by a server in data center is connected with its processor, RAM, hard disk, and bandwidth. Recent studies [22] 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 model function of its present CPU utilization (u) as:

$$P(u) = (P_{idle} * P_{busy}) + (1 - P_{idle}) \times u \qquad (2)$$

where, P is the anticipated power-consumption, P_{busy} and P_{idle} are the power-consumption value once a host is at its maximum utilization state and idle state respectively. The CPU utilization is to be symbolized by u. The utilization of a CPU alters time by time. We define it as a function $u(t)$ of time. As a result, overall energy consumption can be attained by equation 3:

$$Energy = \int_t P(u(t))dt \qquad (3)$$

IV. METHODOLOGY

A. An overview of en BASE (energy efficient network Balanced Algorithm for SLA Ensuring in Dynamic VM Placement)

In this section we are introducing multi- objective combinatorial optimization algorithm named en BASE as a backbone with a Multi Objective ACS (MOBA CoS) for finding global best migration plan for mandatory vm migration to ensuring energy efficiency, reducing migration time as well as down time and reducing SLA violation. Migration is the procedure to shift the VM from the source-host to the destination-host when the former host is either overloaded, prone-to-be overloaded or under loaded. VM migration technique such as live migration is currently proposed by Xen as well as VMware, where the downtime that the customers disseminate owing to the migration procedure is minuscule that ranges in between hundreds of milliseconds to tens of second [23]. Regardless of the advancement of the VM migration methods, it is aggravating yet to expose well-organized resource management schemes that establish which VMs to be migrated and which PMs should be the target for each migration when the present host is relapsed. Hence, a continuous monitoring is essential to identify the diseased PMs and it is very crucial to select a VM from it based on its various features to make the former one in healthy state to ensuing energy efficiency, SLA etc. Moreover, when a VM is to be chosen for migration, it is critical to find out a suitable host in relation with VM characteristics and overall strategy pursued in the data center. enBASE is ensures the mandatory VM migration for avoiding Service Level Agreement Violation which is the collection of three different algorithms such as VMMigration() (for checking whether the VM migration process is done or not), MOBACoS(*Multi Objective Ant Colony System algorithm to find Best Migration Plan to get Destination PM for VM migration), and migPerf()(Live VM Migration Performance Assessment Algorithm).*EnBASE utilizes the benefits of Ant Colony Optimization, which is using the foraging conduct of ants to find food sources from its nests. Here, nests are disguised as diseased PMs from where VM Migration is to be done food sources are masqueraded as Destination PMs.

B. Ant Colony Optimization

In 1997, Dorigo and Gambardella proposed Ant Colony Optimization(ACO) meta heuristic by motivating through the foraging behavior of ants, for resolving the Traveling Salesman Problem (TSP) [24]. In real world scenario, while carrying foodstuff from its originating place to their own nest, the ants drop and go after a chemical essence on their routes named pheromone. It permits them to communicate indirectly with each one to locate enhanced paths among their nest and the source. In ACO, several ants create solutions with the direction of pheromone level and heuristic level information is supplied separately. Here, the pheromone is to be deposited among two cities to document the historical exploration know-how. The heuristic information implanted with a voracious selection according to the present status aids to find an enhanced solution. In every iteration, each ant creates a way by visiting the cities gradually. For the duration of the solution creation, pheromone's local updating is achieved by evaporating the pheromone on the already visited ways and spread out the results. Subsequently, every ants have completed the construction, then the best solution from all the solutions is chosen and used for the pheromone global updating to confirm the best possible solution. The algorithm expires its execution when the iteration limit is reached. A variety of ACO algorithms are there exist. Some of them are Ant System(AS) Max-Min Ant System (MMAS) and Ant Colony System (ACS). Currently ACS [25] is proved one of the top-performing ant algorithm. Hence, in this work, we are applying ACS to the Dynamic Virtual Machine consolidation dilemma.

C. ACS based Dynamic VM Consolidation Problem

VM Migration is the process to shift the VM from the source host to the destination host when a former host is either overloaded or under loaded. In the initial placement of Virtual Machine in each host pm ∈ PM has one or more VMs from the set of Virtual Machines VM. In addition to that, in the milieu of VM migration, each PM should be a probable source PM for the VMs already resides on that PM.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Both them are to be portrayed by their own resource utilizations, such as CPU, RAM, storage and network bandwidth. Similarly, a VM is ready to be migrated to any other PM. Therefore, any PM should be a possible target PM, which is also illustrated by its utilizations of resource which are necessary to include the characteristics of any VMs. Using this algorithm we are finding best migration plan $M_{P_o}^{+}$ to select a suitable destination host P_{dest} for VM migration in accordance with proper routed (i.e., congestion free shortest distance) optimal PMs P_o (average of least energy consumption, highest storage, feasible CPU capacity (processor speed) and superior memory constraints). For the purpose of simplicity, key notions and notations are used in the subsequent segments are tabulravized in Table 1.

Notation	*Meaning*
pm	*Individual Physical Machines*
vm	*Individual Virtual Machines*
PM	*Set of Physical Machines*
VM	*Set of Virtual Machines*
M	*Total number of PMs in a data centre*
N	*Total number of VMs in a particular pm*
P_o	*Set of optimal pms*
P_{src}	*Source pm*
P_{dest}	*Destination pm*
$ovrld_{host}$	*Overloaded host*
$Cndt_{host}$	*Host is prone to be overloaded*
$Undrld_{host}$	*Underloaded host*
P_x	*PMs are excluded as an optimal PM for migration*
T	*Set of tuples*
$M_{P_o}^{+}$	*Global best migration plan*
LMP	*Local migration plan*
MP	*migration plan*
B+S	*Best score achieved by ants*
T	*Set of Tuples*
T_k	*Set of Tuples unvisited by the ant k yet*
S	*Random variable selected according to equation (12)*
η	*Heuristic value*
τ	*The amount of pheromone*
α	*Pheromone decompose constraint in the global update rule*
β	*constraint to settle on the comparative value of* η
q_0	*constraint to decide comparative significance of exploitation*
ρ	*Pheromone decompose constraint in local update ruling*

VM placement and VM consolidation are both time consuming and resource consuming, and are classified as NP-hard optimization dilemma. In this segment, we recommend an algorithm, i.e., MOBACoS used for VM placement scheme that intends to minimize the aggregate energy consumption, resource wastage, and thus improving the Service Level Agreement by avoiding its violation. This algorithm is used to find best migration plan from Local migration plan based on the pheromone trial update. If a migration route has more pheromone than others, that path is suited for migration.

Therefore, the objective function f of the MOBACoS algorithm is:

$$\min f(MP) \coloneqq |P_o| \tag{4}$$

where MP is the Migration Map for all the VMs need to be migrated from the diseased PMs in the data center which is characterized like so:

$$MP_{vm,pm} = \begin{cases} 1, & \textit{if VM vm is to be migrated to PM pm;} \\ 0, & \textit{otherwise} \end{cases} \tag{5}$$

and P_o is the Optimal PM which is the candidate of destination PM. P_o(s) are acquiring some qualities such as the average of certain parameters which is helpful to the smooth organization of data centres, such as shortest distance with congestion free route, least energy consumption, utmost CPU Capacity, good storage level, adequate memory. P_o is the sum total of two algorithms. They are:

$$P_o=\{\ Route\_generate(r[])\,V\ Host\_generate(h[])\} \tag{6}$$

Here, *Route_generate(r[])* is the function which is used to calculate the feasible route by means of shortest distance with minimum bandwidth necessary for congestion free network traffic of migration process of VMs. We found out the shortest paths among every pair of nodes between as well as inside the DC(s) while considering the edge weight as bandwidth. In this circumstance, prerequisite bandwidth of a path has to be over and above the VM's page dirtying rate. The equation for the minimum bandwidth of a certain path ($BW_{m,j}$)is as follows:

$$BW_{m,j} = PD_m(1+\beta) \tag{7}$$

where, m is the present VM migration process , j is the path to be done the migration and PD_m points to the page dirty rate of VM migration at the pre-copy state. β is a minimum hoist with respect of PD_m. It is assumed that 0.3 is the best value through simulation.

Second function *Host_generate(h[])* creates a list of Dest_host[] (Destination host list)from host h[] by applying the following equations by taking host parameters: Energy consumption ,CPU capacity, Memory, Storage.

Find out the mean of aforementioned parameters

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$$\mu_p = \frac{\sum_{i=1}^{n}(Pow)}{n} \quad \text{// Mean of Energy Consumption} \quad (8)$$

$$\mu_{cc} = \frac{\sum_{i=1}^{n}(CPU)}{n} \quad \text{// Mean of CPU capacity (processor speed)} \quad (9)$$

$$\mu_m = \frac{\sum_{i=1}^{n}(RAM)}{n} \quad \text{// Mean of Memory} \quad (10)$$

$$\mu_{ds} = \frac{\sum_{i=1}^{n}(Storage)}{n} \quad \text{// Mean of Disk Storage} \quad (11)$$

As a next pace finding out the deviation from mean for these 4 parameters .

$$p_d = Pow(i) - \mu \quad \text{// for getting least energy consumption} \quad (12)$$

$$cc_d = \mu - CPU(i) \quad \text{// for getting highest CPU capacity (processor speed)} \quad (13)$$

$$m_d = \mu - RAM(i) \quad \text{// for getting highest available memory} \quad (14)$$

$$ds_d = \mu - Storage(i) \quad \text{// for getting highest Storage space} \quad (15)$$

Then, it is finding out the sum of these deviations

$$s[i] = \sum_{i=1}^{n} \int (p_d, cc_d, m_d, ds_d) \text{ // Findout sum of 4 deviation}$$

Sort the sum in descending order for the criteria of selecting Optimal PMs P_o.

Stored them as one element in a tuple t which consists other two elements such as source PM P_{src} and VM to be migrated vm.

The proposed MOBACoS algorithm generates a set of tuples T, where each tuple t ∈ T composes of three aspects: the P_{src} , the migreatble VM vm, and the P_o

$$t = (P_{src}, vm, P_o) \quad (17)$$

Mainly the number of tuples |T| are determining the computational time of the VM consolidation algorithm. Thus, with the aim of reducing the computational time, proposed algorithm is applying some constraints, which produce a a minimized set of tuples by deleting a few least significant and unnecessary tuples. Some constraints given for this concept is follows:

- **Constraint 1:**

It should be necessary to guarantee that only an overloaded($Ovrld_{host}$), a prone to be overloaded - a candidate for becoming an overloaded- ($Cndt_{host}$), or an under-loaded ($Undrld_{host}$) PM is to become a source PM P_{src} from which the VMs should be migrated . The justification of including $Cndt_{host}$ as a P_{src} is to avert the PM from becoming $Ovrld_{host}$. Likewise, the number of SLA violations is to be minimized by migrating a few VMs from an $Ovrld_{host}$. Moreover, migrations from an $Undrld_{host}$ will free up the hosts and set them in the sleep or hibernating mode , which would decrease the energy consumption by reducing the quantity of working PMs.

$$P_{src} \rightarrow P_{src} \in Ovrld_{host} \cup P_{src} \in Cndt_{host} \cup P_{src} \in Undrld_{host}$$

- **Constraint 2:**

Some PMs are excluded - P_x - as destination PMs which are selected for migration based on some criteria that should be satisfied. The P_x are $Ovrld_{host}$, $Cndt_{host}$, and $Undrld_{host}$ which is not to be chosen as destination for migration.

$$P_x \rightarrow P_x \in Ovrld_{host} \cap P_x \in Cndt_{host} \cap P_x \in Undrld_{host} \quad (19)$$

VM is neglecting P_x for migration.

$$P_x := \{\forall P_x \in P | \forall P_x \notin P_{dest} | VM_{P_x} = \emptyset\} \quad (20)$$

- **Constraint 3:**

Destination PM (P_{dest}) is the host which is the candidate for where the migration is need to be happen. Destination PM is not $Ovrld_{host}$ or $Cndt_{host}$ or $Undrld_{host}$. P_{dest} is the member of P_o and not a member of P_x .

$$P_{dest \rightarrow P_{dest} \notin Ovrld_{host} \cap Pdest \notin Cndt_{host} \cap Pdest \notin Undrld_{host}} \quad (21)$$

$$\forall P_o \in P | \forall P_{dest} \in P | \forall P_{dest} \in P_o | \forall P_{dest} \notin P_x \quad (22)$$

Algorithm1: Multi OBjective Ant Colony System algorithm to find Best Migration Plan to get Destination PM for VM migration
Function: MOBACoS(t)

Input: $t = (P_{src}\ vm, P_o)$

Output: Print Best Migration Plan $M_{P_o}^{+}$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

1. *Initialization:* $M_{P_o}^{+} := \emptyset$ *// Global Best Migration plan*
   M := Ø // MigrationMap
   Pheromone value as a criteria t as its first element for each path between Source PM P_{src} *and destination PM* P_{dest}
   ie. $\forall t \in T | \tau t := \tau 0$
   Place N_{ants} *in* P_{src}
2. *Solution generation - s - of each ant:*
   while until a stopping condition is to be met do
   Repeat for each ant
   LMP= NULL//Local migration plan
   MP=NULL//Migration plan
   B⁺S=NULL//best score achieved by ants
   while until LMP check all the tuples of Vertex T do
   Put the ants in the beginning path for Optimal PM from Vertex T i.e., t_0
   for all the remaining path:
   Choose next PM for feasible VM Migration by applying transition rule of $\forall s \in T$

$$Prob_s := \begin{cases} \frac{[\tau_s]^{\alpha}.[\eta_s]^{\beta}}{\sum_{u \in T_k} \{[\tau_u].[\eta_u]^{\beta}}, & if\ s \in T_k \\ 0, & otherwise \end{cases}$$

   Send msg "Optimal PM not found"
   Select a tuple t $\in T$ *to navigate via*

$$s := \begin{cases} \arg\max_{u \in T_k} \{[\tau_u].[\eta_u]^{\beta}\}, & \text{if } q \le q_0 \\ S, & \text{otherwise} \end{cases}$$

   Put the selected Optimal PM P_o *as* P_{dest}
   LMP := LMP $\cup \{t\}$
   End for
   End while
3. *Local Pheromone Updating:*
   Apply local update rule on t using:
   $\tau_s := (1-\rho).\tau_s + \rho.\tau_0$
   if the VM migration in t doesn't overload or prone to be overloaded P_{dest} *then*
   update load capacity vectors
   $LP_{src} = LP_{src} - VM_{ld}$ *// The load of Source is reducing due to the migration of VM*
   $LP_{dest} = LP_{dest} + VM_{ld}$ *//The load of Destination is added with the load of new migrated VM*
   End if
4. *Fitness: Fitness value of ant is: f(LMP)*
5. *Replacement: Substitute the Best score with the ant's solution which is having best fitness value if it has a fitness value that is better than Best Score.*
   if f(LMP) > B⁺S then
   B⁺S:= f(LMP)
   B⁺S:= B⁺S $\cup \{t\}$ *//Best score of ant is added in tuple t*
   end if
   end while
6. *Global Pheromone Updating:*
   M := M $\cup \{MP\}$ *// Migration Map is updated with the migration plan list*
   $M_{P_o}^{+} := \{arg\ maxMP \in M\{f(MP)\}$ *// Accessing the best value for Global Best Migration plan*
   $\tau_s := (1-\alpha).\tau_s + \alpha.\Delta_{\tau_s}^{P_o}$ *// applying global updating rule on every* $s \in T$
   End for
7. *Empty the Vertex T*
8. *Repeat steps 2 to 7 until terminating condition is met. terminating condition may be the utmost limit of iterations or no alteration in fitness value of ants' solutions in successive iterations*
9. *Output: Printing the Best Migration Plan to reach destination PM*

   Return $M_{P_o}^{+}$

In this approach, the pheromone is invested in the tuple t (equation 17) . Each ant makes use of a stochastic-state transition rule to select the subsequent row s from t to traverse for the migration of VM vm to PM P by applying :

$$s := \begin{cases} \arg\max_{u \in T_k} \{[\tau_u].[\eta_u]^{\beta}\}, & \text{if } q \le q_0 \\ S, & \text{otherwise} \end{cases} \tag{23}$$

where , τ_u represents the total quantity of pheromone and η_u stands for the heuristic based value in tuple u of $T_k \subset T$ respectively which are used to construct a finest solution in supported with the optimization function. T_k is a set of tuple that are not to be traversed by the ant k yet. Despite η value, the Pheromone plays a significant part for raising an ant's solution which directs ants to find varied solutions while investigating the search area. α specifies a non -negative random variable which locates the relative importance of pheromone quantity in each path whereas, β denotes a positive constraint to find out the relative significance of the heuristic based value with regard to the pheromone based value provided by the ants in the tuple. q ∈ [0,1] is a consistently distributed random number and 0 ≤ q0 ≤ 1 is a fixed valued input parameter. Besides, S is a random variable preferred in accordance with the probability distribution specified in :

$$Prob_s := \begin{cases} \frac{[\tau_s]^{\alpha}.[\eta_s]^{\beta}}{\sum_{u \in T_k} \{[\tau_u].[\eta_u]^{\beta}}, & if\ s \in T_k \\ 0, & otherwise \end{cases} \tag{24}$$

where, $Prob_s$ denotes the probability of an ant 'k' to chooses tuple '*s*' as next tuple for traversing. The pseudo-random-proportional-rule indicates in (23) and (24) benevolence the tuples with a superior level in concentration of pheromone, which results to find out optimal PMs to disguise as destination PM for VM migration. If q's value is bigger than the value of q0, the search procedure is termed as exploration. Otherwise, it is termed as exploitation.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The input parameter q0 decides the relative significance of exploitation against exploration. In that fashion, if the engendered random number is lesser than q0, the best tuple is preferred that achieves the highest value of $[\tau_u]\cdot[\eta_u]^{\beta}$ according to equation 23 i.e. exploitation , otherwise it is decided with regard to equation 24 i.e. exploration, which chooses the best tuple. The exploitation aids the ants to rapidly congregate to a highest valued solution and it helps to evade stagnation by assigning a broader exploration of the search area.

The heuristic value η_s of a tuple s is defined as:

$$\eta_s \begin{cases} CP_{dest} - LP_{dest} \geq VM_R, & if\ LP_{dest} \leq CP_{dest} \\ 0, & otherwise \end{cases} \quad (25)$$

where, CP_{dest} is the total capacity of the P_{dest}, i.e.;

$$CP_{dest} \leftarrow CPU + BW + RAM \quad (26)$$

In this proposed algorithm we assumed three dimensions of resources such as CPU, Bandwidth and Main Memory. LP_{dest} is the average load of n[th] host is total load of VM divided by number of PM.

$$LP_{dest} = \frac{\sum_{j=1}^{m} VM_j}{n} \quad (27)$$

VM_R is the required resources by VM which is to be migrated in the target PM. In this heuristic function η_s, finding out non overloaded destination for VM migration b checking the constraint

$$LP_{dest} \leq CP_{dest}.$$

In ACS a global as well as a local pheromone trace evaporation rule is using. The local pheromone trace update rule is executed on a tuple when an ant starting its traversal from the very first tuple while building its migration plan.

$$\tau_s := (1-\rho).\tau_s + \rho.\tau_0 \quad (28)$$

where $\rho \in (0,1]$ is the pheromone deteriorate parameter and τ_0 is the primary pheromone point, which is worked out as the multiplicative inverse (reciprocal) of the product of the the approximate most favorable migration plan |MP| and the number of hosts |PM|.

$$\tau_0 = (|MP|.|PM|)^{-1} \text{ OR } \frac{1}{|MP|.|PM|} \quad (29)$$

The global pheromone trace evaporation rule is pertained regarding the end of iteration subsequent to all ants finish their own migration maps. The pheromone level updating by applying global updating rule is defined as follows:

$$\tau_s := (1-\alpha).\tau_s + \alpha.\Delta_{\tau_s}^{P_o} \quad (30)$$

where $\Delta_{\tau_s}^{P_o}$ is the added pheromone quantity which is given only to those tuples that be owned by the global best migration plan so as to provide incentive for them by finding P_{dest} from a minimized number of $P_o(s)$. $\alpha \in (0,1]$ is similar to ρ. It is termed as:

$$\Delta_{\tau_s}^{P_o} := \begin{cases} |P_o|, if\ s \in M_{P_o}^{+} \\ 0, \quad otherwise \end{cases} \quad (31)$$

Here, $M_{P_o}^{+}$ is the global best migration plan is getting achieved from the commencement of the traversal made by the ants.

Algorithm2: Checking the VM migration process

VmMigration(host h[], route r[],VM v)
Call Route_generate(r[])
Call Host_generate(h[])
if Finest_host and Finest_ route then
return 1
else
return 0

This algorithm is checking whether the migration process is completed or not by calling two functions *Route_generate(r[])* and *Host_generate(h[])* and the return value of the aforementioned functions Finest_host and Finest_route is respectevely . If both the values are true, it is assumed that migration is completed successfully. Otherwise, there is a failure of migration process.

An energy-efficient network Balanced Algorithm for SLA Ensuring in Dynamic VM Placement (enBASE)

Algorithm3: Energy efficient network balanced algorithm for SLA Ensuring in Dynamic VM Placement

enBASE (Host h [],VMList vm,n,m)

Input: I(P_{so},VM,P_o[])
Output: VMPlacementmap
Interface with cloud data
VM is designed and allocated to hosts;
Workload is assigned to VM;

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

For each period t;
Identify overloaded and under loaded host - source_PM- by threshold;
Pick the most suitable VM based on its highest CPU usage and highest response time;
Consider the least power consumed hosts;

If host h is inside the data centre then
VM is assigned to h;
Else
Finest_host<- MOBACoS(I) //Algorithm 1
Place the VM in Finest_host;
End if

If VmMigration()==1 then //Algorithm2
Return VMPlacementmap
Remove vm from Source_host
Exit
Else
Remove Finest_host
call enBASE;
End if

End for

Here, enBASE is identifies the diseased PMs by the constant monitoring of the system. It selects VMs based on its highest CPU utilization as well as its highest response time. It is already stated in our prior work[5]. At the initial part of this algorithm, it is assigning the VM in the destination host based on the Multi objective ACS algorithm MOBACoS. In the second part, it is making a continuous monitoring of the travel of VM from source to destination until it is completely migrated from the former to latter and resume its task in the latter. Then only it is completely removing from the source host. If migration is not to be done it is recursively checking the target host with aforementioned host selection criteria along with congestion free streaming network traffic.

D. Performance Metrics

a.. Energy consumption

A data centre's energy consumption counts the total energy consumed by different elements of the data center such as CPU, memory, storage, energy supply components and air cooling arrangements.

$$Energy = \int_t P(u(t))dt \qquad (3)$$

The detail description is shown in section III.

b. Total Migration time

This is the time period in use of entire transfer of virtual machine from source PM to target PM. It should be lowest time for rapid completion of migration.

$$T_{mig} \leftarrow \sum_{i=0}^{n} T_i + T_{ist} \qquad (32)$$

where, T_{mig} is the total migration time which is the aggregate of time duration taken by all the iterations for relocating the entire virtual machine from source to destination.

c. Downtime

This is the time duration in use of migration procedure to stop the execution of virtual machine at source host and resume the operations at target host.

$$T_{dwn} \leftarrow T_{ist} + T_{resume} \qquad (33)$$

d. SLA Violation Time per Active Host (SLATAH)

Table 1: The number of VMs in the real work load (bit Brains)

Date	No of VMs
12th August, 2013	1228
16th August, 2013	1223
19th August, 2013	1214

The proportion of time in which active servers occupied 100% (Complete) CPU or memory utilization

$$\text{SLATAH}= \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}} \qquad (34)$$

where N is the total number of PMs; T_{si} is the aggregate time that the i^{th} PM has been practiced the CPU or memory utilization of 100% which are bringing out an SLA violation. T_{ai} is the total of the i^{th} PM being in to the running condition.

e. Performance Degradation due to Migration (PDM)

The entire performance degradation by VMs caused by live migration on another physical host in network level. It measures the whole SLA violation due to live VM migration. When a host is overloaded, needed VMs are migrated from that host to a normal-loaded host. In another condition, when a host is underloaded, all the VMs are migrated from this diseased host to another normal-loaded host. At the duration of migration, that VM is not competent of serving user requirements. Therefore, it grounds SLA violation. This metric computes the SLA violation originated by migration. PDM can be helped to achieve the selection of a migration technique that gains less load on the network.

$$\text{PDM}= \frac{1}{M}\sum_{j=1}^{M}\frac{Cd_j}{Cr_j} \qquad (35)$$

where, parameter M signifies the total number of VMs; parameter Cdj C_{dj} is the assessment of performance degradation due to live migration of j^{th} VM (VM_j); constraint Cr_j denotes that the total CPU capacity required by VM_j throughout its life span. Given that both SLATAH and PDM metrics are of equally vital for independently measuring the SLA violations.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-2S, December 2018

f. Service Level Agreement Violation(SLAV)

To confine effect of the above mentioned two metrics SLATAH and PDM on SLA violations the joined metric SLAV is calculated.

$$SLAV = SLATAH \times PDM \qquad (36)$$

V.RESULTS & PERFORMANCE APPRAISAL

A. Workload Data

In order to assess the proposed algorithm enBASE on real workload data sets and to be capable to redeem genuine outcomes, we exercised Grid Workloads Archive (GWA), GWA-T-12 Bitbrains traces from Delft University (http://gwa.ewi.tudelft.nl) as experimental data [26]. It is conducted with a set of different number of VMs from 3 random dates which were selected from the obtained traffic traces. We occupy the equivalent power models supplied in the website for both servers as publicized in the following table:

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

Table 2: The power consumption at dissimilar load levels in Watts

The specifications of PMs and VMs which are used in the works are listing in Table 2 and Table3, correspondingly.

Table 3: Host Parameters

Parameters	Server	
	HP ProliantG4	HP ProliantG5
No of host	400	400
No of Cores	2	2
MIPS	1860	2660
RAM	4096	4096
BW	1GB	1GB
Storage	1.5GB	2GB

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	1MB	1MB	1MB	1MB
Storage	3.85GB	2GB	1.75GB	613MB

VI.EXPERIMENTAL SETUP

We have proposed a large-scale IaaS environment which provides incredible 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 ever more popular due to their assistance for elastic, scalable, proficient and repeated evaluation of resource provisioning procedures for various applications[27]. Bitbrains expose a data set enclosing workload traces for 1750 VMs from its distributed datacenter standing for hosting and business-critical venture applications. The trace is consisted of one file per VM depicting generally the VM's dynamic workload, trialed in each 5 minutes. This workload file was interpreted by CloudSim to generate various number of cloudlets. These data comprised of the used CPU and main memory of the VM at the certain period of time. It also comprised of disk and network I/O throughput values. As the traffic rate matrix, the traffic rates are unforeseeable, and albeit they are diverge among VMs from time to time.

The network traffic data utilized in this effort is attained through III party source, which are to be availed the data accessible for public. The 3 authentic data center traffic traces are issued by the website of University of Wisconsin-Madison(http://pages.cs.wisc.edu/tbenson/IMC10_Data.htm l.) [28]. In this work, we are selected University data centre – UNI1 data center traces for the data center traffic experimental simulation. The beginning timestamp of the data exploited in this work is 00:00:00 and the ending timestamp is 23:59:59 of three random dates. This data imitate the traffic nature of a typical data center overlonger time periods and can be took a broad view for other data centers also. We confirm our topology as fat-tree, where each physical machine has equal resource capacity.

To check the efficiency of our approach we compare enBASE with the support of MOBACos an ACS based algorithm, with two ACS based SLA oriented Energy Efficient VM consolidation algorithms, such as UACS, ACS-VMC and the traditional heuristic approach, BFD algorithm named MST. Best First Decreasing is widely adopted for Energy Efficient VM consolidation in bin packing model and is used as a criterion. It deals out the VM to the host that fit best. Ant Colony System (ACS), meta-heuristic is instigated from the foraging behavior of existent ant colonies. VM consolidation use this highly adaptive meta-heuristic algorithm. For MOBACos the associated parameters are $q_0 = 0.8$, $\rho = 0.1$, $\alpha = 0.2$, $\beta = 1.0$, number of ants $N_{Ants} = 10$ and number of iteration $N_{itn} = 20$.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

A novel ACS-based, Multi Objective Ant Colony System(MOBACoS)algorithm is suggest as a solution to the proposed enBASE algorithm. The MOBACoS algorithm is used to find global best migration plan for mandatory migration executed in enBASE to reduce SLA violation in connection with less energy consumption, minimization of total migration time as well as downtime.

MST algorithm [29] is concentrating on over-utilized host and avoiding network-congestion. But they are completely ignoring under-utilized host and prone to be over-utilized host. They are selecting the VMs from congested links and not selecting based on load of the servers. Due to the ignorance of base reason(ie., the overloaded hosts), it is not much gaining energy. It selects a single VM from congested link. It is not ensuring the full non congestion facility within a limited time. So it is the chance of increase of migration time as well as downtime. Moreover, they are considering predefined static threshold value for finding upper load of servers. Thus it is not suitable for varying workload which is not predictable. This is affecting the improvement of SLA because based on the varying workloads , the algorithm is not be able to migrate VM from the overloaded host(s).

ACS-VMC algorithm[7] is formulating energy-efficient VM consolidation as a multi-objective combinatorial optimization problem to optimize three contradicting objectives simultaneously. The objectives are including reduction of energy consumption, minimization of the number of VM migrations, and avoidance of SLA violations. But in ACS-VMC, a migration is permitted only if it be able to decrease the lively host number and the quantity of migrations. This selection technique creates a circumstance in which the VMs scheduled in overloaded hosts can't be migrated and thus causes to a lesser quantity for energy consumption, but higher level of SLA violation. This indicates that ACS-VMC often locates solutions that breach the SLA constraints, which are not competent in real world cloud structures.

UACS algorithm[8] is selecting a sever regarding the CPU capacity and energy consumption in both overloaded and underloaded situation for VM migration. While considering the load of the server, they are deemed only static threshold method . Such that in excess of 90% total amount of available CPU utilization is taken as overloaded PM and under 50% total amount of available CPU utilization is considered as underloaded PM. So it is not suitable for real time workload fluctuations. They consider only two types of resources: CPU and RAM. They completely ignoring the network routing and bandwidth which has a key role in live VM This is the cause of total migration time and down time increase.

Surely, this will increase the violation of SLA at a great extent.

Workload	Algorithm	Energy Consumption (kWh)		Total Migration Time (ms)		Down Time (ms)		SLA Violation (%)	
		Value	Comparison (%)	Value	Comparison (%)	Value	Comparison (%)	Value	Comparison (%)
12th August,2013	MST	115.5	25.9	198	15.7	152	40.8	3.64	33.8
	UACS	100.5	14.8	171	2.3	124	27.4	3.44	29.9
	ACS-VMC	96.8	11.6	213	21.6	91	1.1	3.21	24.9
	enBASE	85.6		167		90		2.41	
16th August,2013	MST	113.4	18.6	215	29.8	159	42.1	3.62	32.9
	UACS	112.6	18	182	17	124	25.8	3.4	28.5
	ACS-VMC	110.2	16.2	223	32.3	127	27.6	3.21	24.3
	enBASE	92.3		151		92		2.43	
19th August,2013	MST	120.1	21.4	212	28.3	118	14.4	3.63	33.6
	UACS	97.6	3.3	213	28.6	120	15.8	3.39	28.9
	ACS-VMC	101.9	7.4	180	15.6	109	7.3	3.22	25.2
	enBASE	94.4		152		101		2.41	
Average	MST	116.3	21.9	208	24.5	143	34.3	3.64	33.5
	UACS	103.6	12.4	189	16.9	123	23.6	3.41	29
	ACS-VMC	103	11.8	205	23.4	109	13.8	3.21	24.6
	enBASE	90.8		157		94		2.42	

Table 5: Simulation results and comparison value of enBASE with benchmark algorithms (mean values)

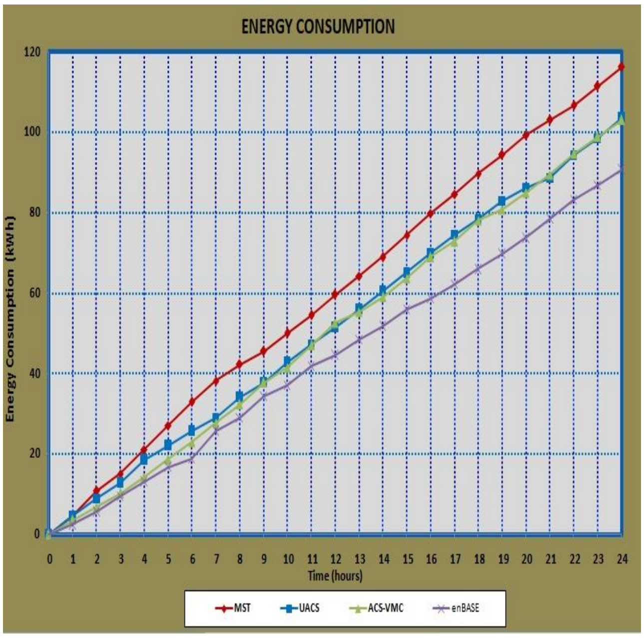

Figure 3: Energy consumption by enBASE and other benchmark scheme for random workload traces (mean values)

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

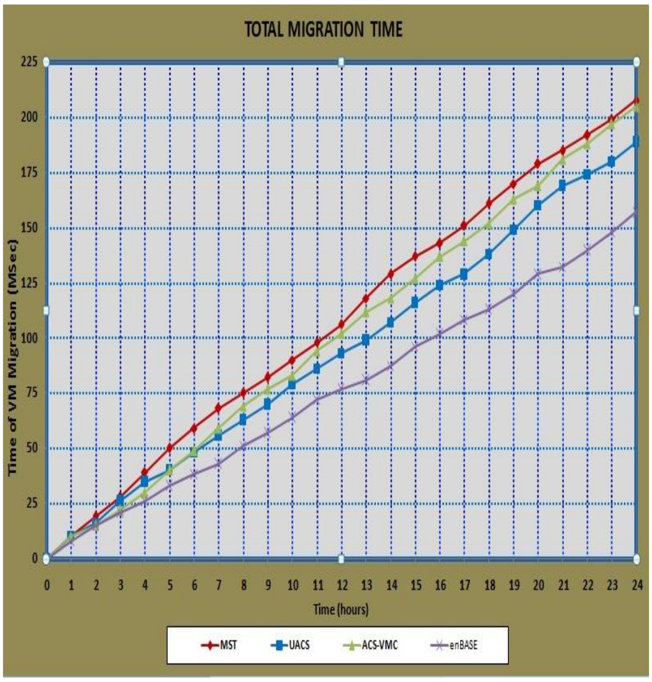

Figure 4: Total Migration Time by enBASE and other benchmark scheme for random workload traces (mean values)

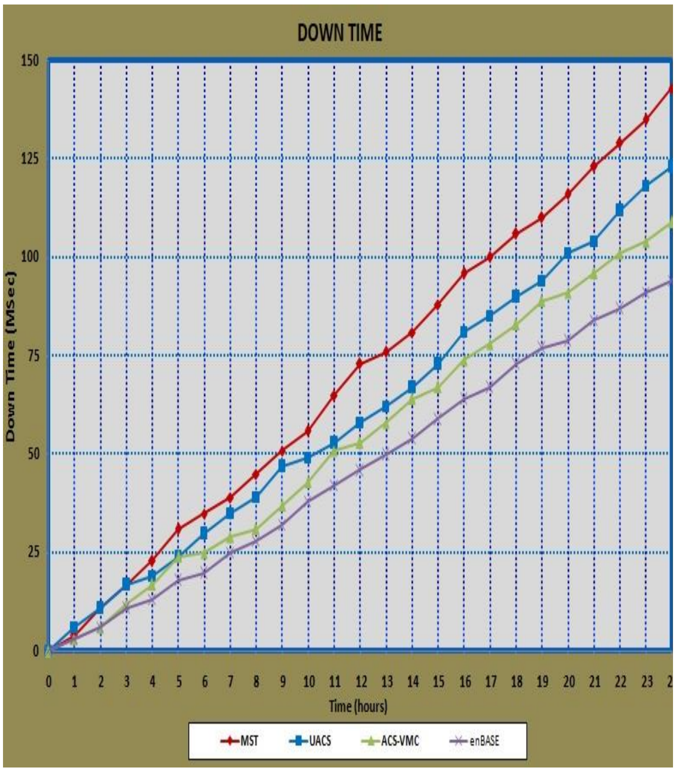

Figure 5: Down Time by enBASE and other benchmark scheme for random workload traces (mean values)

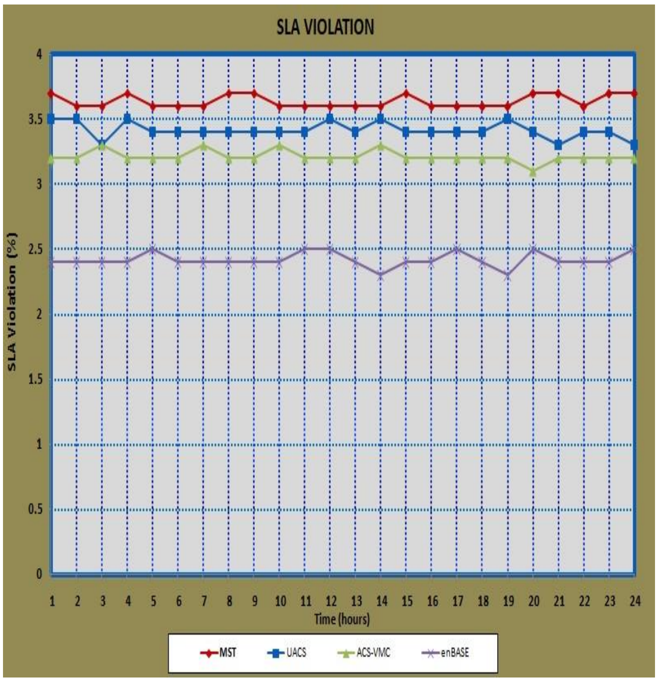

Figure 6: SLA Violation by enBASE and other benchmark scheme for random workload traces (mean values)

The proposed algorithm enBASE is a combinatorial optimization problem by finding an optimal object from a finite set of objects, which ensures the mandatory VM migration in optimal host with multiple criteria such as shortest distance in conjunction with proper bandwidth route to reach the energy efficient sufficient resource and highest CPU capacity in terms of processor speed enabling host. This is improving the energy efficiency and reducing the total migration time as well as downtime. Consequently, it is reducing the SLA violation. In enBASE, a considerable reduction of the average power consumption of 11.6%, 14.8%, 25.9% be attained as compared to ACS-VMC, UACS, and MST respectively. Moreover, in enBASE, a major decreasing of the average SLAV of 24.9%, 29.9%, and 33.8% was acquired as compared to ACS-VMC, UACS, and MST correspondingly. It was also shown good performance result (shown in Table 5) in the reduction of total migration time and down time while comparing afore mentioned algorithms. Thus, it is proved through the simulation that it outperforms all above mentioned existing algorithms.

VII.CONCLUSION

In this paper, we suggested an energy- efficient network-oriented SLA aware Dynamic VM Consolidation algorithm termed enBASE with a backbone support of Multi Objective ACS algorithm named MOBACoS. It diminishes the energy using up of data centres by consolidating VMs in a minimized number of PMs in a congestion free network route. It ensures the mandatory VM migration in destination hosts, thus it markedly reduce the SLA violations.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

We exploited the eminence of Ant Colony System (ACS) to discover a nearly-optimal solution with least energy consumption and provide better support for SLA, by seeing as the VM consolidation problem is treated as strictly NP-hard. While finding out a minimized number of optimal PMs with some qualified constraints to become a destination PM, which results the decline in energy consumption, less number of migrations, reduction in total migration time along with downtime. As a byproduct of the excellence of this algorithm, SLA violation can be considerably reduced and the Quality of Service maintained. The simulation result showed that while comparing with the existing dynamic VM consolidation approaches such as BFD and ACS , enBASE surpassed in aforementioned objectives.

As a future work, planning to extend the approach with improvements and additional functionalities to ensure the security in VM migration and placement.

REFERENCES:

1. Namasudra, S. (2018). CLOUD COMPUTING: A NEW ERA. Journal of Fundamental and Applied Sciences, 10(2), 113- 135.
2. EC-European Commission. (2007). Limiting Global Climate Change to 2 degrees Celsius. The way ahead for 2020 and beyond. COM/2007/2. (Accessed in July 2018).
3. Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2010). Cloud computing: Principles and paradigms (Vol. 87). John Wiley & Sons.ISBN: 978-0-47088799-8
4. SLA, C. (2014). Cloud service level agreement standardisation guidelines. European Commission, Brussels.
5. Sajitha, A.V, & Subhajini, A.C (2018). Dynamic VM Consolidation Enhancement for Designing and Evaluation of Energy Efficiency in Green Data Centers Using Regression Analysis. *International Journal of Engineering & Technology*, *7*(3.6), 179-186. doi:http://dx.doi.org/10.14419/ijet.v7i3.6.14966.
6. Ferdaus, M. H., Murshed, M., Calheiros, R. N., & Buyya, R. (2017). Multi-objective, Decentralized Dynamic Virtual Machine Consolidation using ACO Metaheuristic in Computing Clouds. arXiv preprint arXiv:1706.06646.
7. Farahnakian, F., Ashraf, A., Pahikkala, T., Liljeberg, P., Plosila, J., Porres, I., & Tenhunen, H. (2015). Using ant colony system to consolidate VMs for green cloud computing. IEEE Transactions on Services Computing, 8(2), 187-198.
8. Liu, X. F., Zhan, Z. H., & Zhang, J. (2017). An Energy Aware Unified Ant Colony System for Dynamic Virtual Machine Placement in Cloud Computing. Energies, 10(5), 609.
9. Ashraf, A., & Porres, I. (2018). Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system. *International Journal of Parallel, Emergent and Distributed Systems*, *33*(1), 103-120.
10. Aryania, A., Aghdasi, H. S., & Khanli, L. M. (2018). Energy-Aware Virtual Machine Consolidation Algorithm Based on Ant Colony System. *Journal of Grid Computing*, 1-15.
11. Hassan, M. K., Babiker, A., Baker, M., & Hamad, M. (2018). SLA Management For Virtual Machine Live Migration Using Machine Learning with Modified Kernel and Statistical Approach. Engineering, Technology & Applied Science Research, 8(1), 2459-2463.
    Mustafa, S., Bilal, K., Malik, S. U. R., & Madani, S. A. (2018). SLA-Aware Energy Efficient Resource Management for Cloud Environments. IEEE Access, 6, 15004-15020.
12. Su, W., Hu, J., Lin, C., & Shen, S. (2015, June). SLA-aware tenant placement and dynamic resource provision in SaaS. In Web Services (ICWS), 2015 IEEE International Conference on (pp. 615-622). IEEE.
13. Anan, M., & Nasser, N. (2015, December). SLA-Based Optimization of Energy Efficiency for Green Cloud Computing. In Global Communications Conference (GLOBECOM), 2015 IEEE (pp. 1-6). IEEE.
14. Zhang, J., He, Z., Huang, H., Wang, X., Gu, C., & Zhang, L. (2014, December). SLA aware cost efficient virtual machines placement in cloud computing. In Performance Computing and Communications Conference (IPCCC), 2014 IEEE International (pp. 1-8). IEEE.
15. Jungmin, S., Amir V.D., Rodrigo N. C. and Rajkumar B.(2017) "SLA-aware and Energy- Efficient Dynamic Overbooking in SDN-based Cloud Data Centers" ,IEEE Transactions on Sustainable Computing Volume: 2, Issue: 2. [11] Nguyen, T. H., Di Francesco, M., and Yla-Jaaski, A. (2017). Virtual Machine Consolidation with Multiple Usage Prediction for Energy- Efficient Cloud Data Centers. IEEE Transactions on Services Computing.
16. Nguyen, T. H., Di Francesco, M., and Yla-Jaaski, A. (2017). Virtual Machine Consolidation with Multiple Usage Prediction for Energy- Efficient Cloud Data Centers. IEEE Transactions on Services Computing.
17. Mohammad, A.K., Mohd, N.D., Azizol, A., Shamala, S., and Mohamed, O.(2017) "Energy- Efficient Algorithms for Dynamic Virtual Machine Consolidation in Cloud Data Centers", IEEE Transactions on Green Cloud and Fog ComputingEnergy Efficient and Sustainable Infrastructures, Protocols and Applications Vol.5 , Issue: 69 PP no: 10709 – 10722.
18. Justafort, V. D., Beaubrun, R., & Pierre, S. (2015). On the carbon footprint optimization in an intercloud environment. IEEE Transactions on Cloud Computing.
19. F. Farahnakian, T. Pahikkala, P.Liljeberg,J.Plosila, N.TrungHieu, and H.Tenhunen, "Energy- aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model", IEEE Transactions on Cloud Computing Volume: PP, Issue: 99 , 2016
20. Zhou, A., Wang, S., Hsu, C. H., Kim, M. H., & Wong, K. S. (2017). Network failure–aware redundant virtual machine placement in a cloud data center. Concurrency and Computation: Practice and Experience, 29(24).
21. Beloglazov, A., & Buyya, R. (2013). Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. IEEE Transactions on Parallel and Distributed Systems, 24(7), 1366-1379.
22. Dabbagh, M., Hamdaoui, B., Guizani, M., & Rayes, A. (2016). An energy-efficient VM prediction and migration framework for overcommitted clouds. IEEE Transactions on Cloud Computing.
23. Dorigo, M. & Gambardella, L.M., (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem, Evolutionary Computation, IEEE Transactions on 1, pp. 53–66.
24. Mohapatra, S., Mohanty, S., Pattanayak, S., & Hota, A. (2017). Comparison of Various Platforms in Cloud Computing. International Journal of Computer Applications, 162(7).
25. Ghasemi, S., Meybodi, M. R., Fooladi, M. D. T., & Rahmani, A. M. (2017). A cost-aware mechanism for optimized resource provisioning in cloud computng. Cluster Computing, 1-14.
26. Shere, R., Shrivastava, S., & Pateriya, R. K. (2017). CloudSim Framework for Federation of identity management in Cloud Computing.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

27. Tajiki, M. M., Akbari, B., & Mokari, N. (2017). Optimal Qos-aware network reconfiguration in software defined cloud data centers. Computer Networks, 120, 71-86.
28. Borhani, A. H., Hung, T., Lee, B. S., Qin, Z., & Bagheri, Z. (2017, March). Network-Aware VM Migration Heuristics for Improving the SLA Violation of Multi-Tier Web Applications in the Cloud. In *Parallel, Distributed and Network-based Processing (PDP), 2017 25th Euromicro International Conference on* (pp. 454-462). IEEE.
29. Sajitha, A.V, & Subhajini, A.C (2018, October).. Network-Aware VM Migration for Achieving Energy Efficient Green Data Centres through Dynamic VM Consolidation, Journal of Advanced Research in Dynamical and Control Systems 10(8) (pp 361-375)ISSN 1943-023X.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)