BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 2
Sofia • 2016 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0023

Evolutionary Computing Based on QoS Oriented Energy Efficient VM Consolidation Scheme for Large Scale Cloud Data Centers

Perla Ravi Theja[1], *S. K. Khadar Babu*[2]

[1]*School of Computer Science & Engineering, Research Scholar, VIT University, Vellore, India*
[2]*School of Advanced Sciences, VIT University, Vellore, India*
Emails: ravithejaperla9048@gmail.com khadar.babu36@gmail.com

Abstract: *The high pace and increase in cloud computing technology and associated applications, especially large scale data centres, have demanded energy efficient and Quality of Service (QoS) oriented computing platform. To meet these requirements, virtualization and Virtual Machine (VM) consolidation has emerged as an effective solution. The optimization in VM consolidation by means of efficient dynamic resource-utilization prediction, VM selection and placement can achieve optimal solution for energy efficient and QoS oriented cloud computing system. In this paper, an evolutionary computing algorithm called Adaptive Genetic Algorithm (A-GA) based VM consolidation approach has been developed. A-GA based placement policy and its implementation with different VM selection policies like Minimum Migration Time (MMT), Maximum Correlation (MC) and Random Selection (RS), along with different CPU utilization estimation approaches like Inter Quartile Range (IQR), Local Regression (LR), Local Robust Regression (LRR), static THReshold (THR) and Median Absolute Deviation (MAD) has revealed that A-GA based consolidation with MMT selection policy and combined IQR and LRR can enable optimal VM consolidation for large scale infrastructures. In addition, the proposed A-GA policy has exhibited better performance as compared to other meta-heuristics such as Ant Colony Optimization (ACO) and Best Fit Decreasing. The proposed consolidation system can be used for large scale cloud infrastructures where energy conservation, minimal Service Level Agreement (SLA) violation and QoS assurance is inevitable.*

Keywords: *Energy efficient cloud computing, virtual machine consolidation, adaptive genetic algorithm, minimum migration time, dynamic threshold, service level agreement violation.*

1. Introduction

The rising demand for cloud infrastructures has motivated entrepreneurs and organizations to establish large scale data centers that comprise thousands of computational nodes or Physical Machines (PMs), also called host nodes, thus causing enormous energy consumption. This demanding energy consumption can significantly impact various factors like environmental issues, resource sustainability, cost of service, etc. Hence, there is an inevitable need to reduce energy consumption. In addition, the exponentially increased number of users for cloud computing services demands, the Quality of Service (QoS) and service reliability. To deal with these issues and ensure optimal resource utilization, energy conservation and QoS provisioning, virtualization has emerged as a potential solution. Virtualization performs live migration of Virtual Machines (VMs) across the network while ensuring zero or minimal downtime. Live migration concept is of great significance, especially for VM consolidation, as it can enable minimum active hosts, thus resulting into reduced energy consumption. The persistent VM consolidation can cause performance degradation due to dynamic resource requirements. Assuring QoS services and its reliability as per Service Level Agreements (SLA) is an unavoidable need for cloud service providers. To achieve these objectives, VM consolidation has emerged as a potential solution. Dynamic VM consolidation [1] can be significant for leveraging the fine-grained variations in the workload, thus enabling minimal active host's count, which might significantly reduce energy consumption. The dynamic VM consolidation approach encompasses two elementary processes. First the VM migration from certain underutilized host or PM and second offloading the VMs from hosts whenever it undergoes overloading situation so as to reduce any possible performance degradation. In first case, the idle nodes or nodes undergoing underutilization are SWITCHED OFF so as to reduce the active or static power consumption. Further, as per network needs, the host nodes are reactivated to accommodate migrated VMs.

In general, the overall VM consolidation scheme encompasses four main phases. The first and second phases deal with under load and overload detection respectively, while it is followed by VM selection in the third phase that decides which VMs are to migrate. Finally, in the last phase of consolidation, the VM placement takes place where these VMs are allocated while ensure that it wouldn't cause overload on the host. Considering dynamic functional environment, the dynamic resource allocation and scheduling can be of great significance. Meanwhile, the efficient VM selection and placement can play vital role for enhancing overall performance of cloud infrastructure. On the contrary, the inefficient VM selection and allocation might result in severe performance degradation and SLA violation issues. To perform efficient VM selection, the statistical analysis of VM resource requirements and its behaviour mapping should be considered. Normally, VM consolidation is a bin packing problem that can be effectively dealt with by means of certain heuristic approaches and evolutionary computing schemes. Considering these motivations, in this paper, we have proposed a highly robust and efficient VM consolidation scheme where it has been intended to enhance overall functional components such as, dynamic threshold based overloading detection, enhanced VM selection policy and evolutionary computing based VM placement facility. In this paper an enhanced dynamic threshold estimation scheme based on scheduling based combined Local Robust Regression (LRR) and Inter Quartile Range (IQR) algorithm has been developed, which has been followed by VM selection for which three different algorithms, Minimum Migration Time (MMT), Maximum Correlation

(MC) and Random Selection (RS) policy have been used independently. Finally, to enhance VM placement, we have proposed and developed two algorithms namely A-GA (Adaptive Genetic Algorithm) and ACO (Ant Colony Optimization) scheme. The results obtained for different combinations reveal that A-GA with MMT selection policy performs better in terms of energy consumption, SLA Violation (SLAV), etc.

2. Related work

A significant amount of research has explored VM consolidation approaches [2-5] for QoS and energy optimization. To deal with overload detection in data centers, initially researchers used the static threshold approach [6], where they considered overall CPU utilization and scheduled hosts to be in the range of certain defined thresholds. Later, considering highly dynamic cloud environment, the conventional static threshold based scheduling was found ineffective to alleviate SLA and therefore the dynamic cloud pattern and workloads based threshold schemes were proposed [7]. Buyya and others authors (see [7]) employed an adaptive threshold scheme for upper and lower bound estimation, which was defined on the basis of the statistical analysis and historical pattern of CPU utilization. Regression based CPU utilization was suggested in [8, 9], where the CPU utilization was estimated at host nodes. Researchers used linear regression and the K-nearest neighbour regression schemes for approximating the data retrieved throughout the VMs lifetime. They emphasized their model for SLA optimization. Later, bin packing problem was considered in [10-12] for efficient VM consolidation. Majority of existing approaches have used Best Fit Decrease (BFD) based consolidation [2]. The algorithms made effort to minimize number of host count by packing more and more VMs onto host. Some combinatorial optimization mechanism were proposed by researchers where they tried to incorporate adaptive schemes for consolidation optimization [13, 14, 15].

Researchers suggested a bio-inspired technique [16] to reduce energy consumption among servers. The foraging of ant based resource allocation was proposed in [17]. Some heuristic approaches like ACO and evolutionary concept of Genetic Algorithm (GA) were suggested in [18] and [19] respectively, for consolidation purposes. ACO-based decentralized scheme for scheduling was proposed in [20]. The genetic algorithm was examined for resource scheduling function in virtualization [21-23]. The majority of existing research has either focused on VM selection or VM placement. Some of the researchers have made efforts towards introducing dynamic thresholding to provide optimistic resource allocation facility [7].

3. Our contribution

In this paper, we performed a comparative analysis between different combinations of host overload detection algorithms, VM selection policies and VM placement policies. We came to see our proposed, A-GA based consolidation with MMT selection policy and combined IQR and LRR can enable optimal VM consolidation, with minimal SLA violation, QoS assurance and provides energy conservation for large scale infrastructures.

3.1. Cloud model

In this paper, we have considered a large scale cloud infrastructure comprising N heterogeneous host nodes or PMs, where each host is characterized in terms of its CPU utilization and Millions Instructions Per Second (MIPS). As, it doesn't have its own inherent storage capacity, a Storage Area Network (SAN) has been used to support live migration. In addition, we assume that there is no available information about the application, execution time or associated workloads for which the migration has to be done. It states that our approach is the application-agnostic paradigm. In this model, multiple independent VMs requests for different MIPS. Thus, the VM consolidation on a single host is the problem of resource management when the combined workload caused due to individual VM and associated applications demands resource simultaneously.

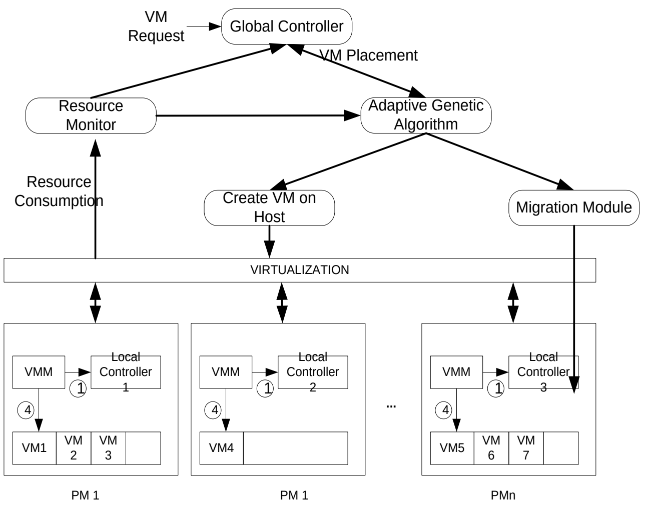

Fig. 1. Proposed system architecture

The VM migration from one host to another might facilitate better resource allocation and energy conservation, but during this process it might suffer certain downtime, causing SLA violation. For service provider, meeting SLA and QoS reliability are the inevitable requirements. Here, we consider that efficient resource allocation, VM migration, optimal VM selection and its swift placement across the network can achieve optimal performance, meeting overall expectations. In the proposed system, a multi-controller based VM consolidation scheme has been proposed. We have estimated host underload and overload locally on each host node. Furthermore, the VM placement has been done on the individual host. The proposed system encompasses two controllers; Global Controller (GC) and Local Controller (LC) Fig. 1. LC functions as Virtual Machine Manager (VMM) for each node and monitors resource (CPU) utilization continuously and identifies the underload and overloaded nodes during execution. Recognizing overloaded node, the LC initiates

VM selection scheme to find which specific VM to offload and from which host, so as to avoid overloading. GS on the other hand retrieves information from LC to enable dynamic scheduling. Based on the decision of the LC, the GC generates migration command to perform VM placement. During this process, VMM executes VM migration and accordingly updates the changes in power modes at different hosts to enable further scheduling and placement.

3.2. Host under load detection

In our proposed system, if a host node is undergoing minimal resource utilization, then it is considered an under load host and therefore all of its connected VMs are migrated from it to other host(s), without causing overloaded on them. Performing VM migration, to conserve energy the source host node is either SWITCHED OFF or in SLEEP MODE. In our approach, if all VMs from the host node couldn't be offloaded, then the source node is still active so as to alleviate the probability of downtime and QoS degradation.

3.2.1. Host over load detection

To detect the overloaded host node, each host node initializes an overload detection scheme intermittently to perform VM de-consolidation, thus enabling SLA violation avoidance. CPU utilization of the host node has been used for identifying overloaded hosts. Unlike conventional schemes, based on static threshold, in this paper we have developed a dynamic threshold based adaptive CPU utilization and overload detection scheme. It enables the proposed system to behave in real time scenario where there is highly fluctuating resource utilization. It adjusts resource utilization threshold based on the variation in CPU utilization and utility map. It can be observed that higher deviation might even result into 100% CPU utilization that signifies higher overloading probability. To enable dynamic threshold detection scheme, we have used IQR and LRR algorithm. In our proposed model, the resource utilization has been examined at the interval of 5 min and on each odd iteration, IQR algorithm has been used, while LRR has been scheduled for even iterations. Such novelty tends to employ major advantages of both approaches. A brief of IQR is given as follows:

IQR is a statistical dispersion technique which is equivalent to the differences between the third and first quartile. To estimate dynamic CPU utilization threshold, the following equation has been used:

$$\mathrm{IQR} = \mathrm{Q}_3 - \mathrm{Q}_1, \quad \mathrm{T}_\mathrm{m} = 1 - \mathrm{s}.\mathrm{IQR}, \tag{1}$$

where s represents the safety parameter that states the maximum extent of the tolerability of a host node in Cloud environment and its lower value signifies the higher tolerance to the fluctuation in the CPU utilization. Here we have used $s = 1.2$ and it can be changed to examine the optimal performance.

In addition to IQR, we have used the Loess concept [24] to derive the LRR algorithm, which has been employed for fitting a trend polynomial to the earlier k observations for the CPU utilization called utilization map. For initial recent observations, it is retrieved as

$$\hat{g}(x) = \hat{a} + \hat{b}x. \tag{2}$$

It is further used for calculating the next observation $\hat{y}(x_{k+1})$. To offload some VMs from an overloaded host node, the following conditions have been fulfilled:

(3) $$s.\hat{y}(x_{k+1}) \geq 1, x_{k+1} - x_k \leq t_m,$$

where $s \in \mathbb{R}^+$is the safety parameter that signifies the maximum capability or tolerability of a host node, and t_m represents the maximum time required for migrating a VM from the overloaded host. The traditional Loess concept [24] is found vulnerable to the outliers caused due to leptokurtic or heavy-tailed distributions. To alleviate this problem, we have modified the Loess concept [24] to bisquare from the conventional least-squares LR approach. Thus we have LRR approach iteratively for estimating the initial fitting and tricube weight function has been used for dynamic weight estimation. The fitting parameter has been retrieved at x_i to get optimal value by means of $\hat{y}_i$. Thus, the final residual value is $\varepsilon_i = y_i - \hat{y}_i$. Furthermore, the final retrieved value (x_i, y_i) has been assigned a robustness factor $\mathcal{R}_i$ that primarily depends on the magnitude of ε_i. Mathematically, $\mathcal{R}_i$ is obtained as

(4) $$\mathcal{R}_i = \mathcal{B}\left(\frac{\hat{\varepsilon}_1}{6L_{\text{MAD}}}\right),$$

where $\mathcal{B}(.)$ gives the bisquare weight function and L_{MAD} represents the Median Absolute Deviation (MAD) for the least square fit. Mathematically:

(5) $$\mathcal{B}(.) = \begin{cases}(1 - u^2)^2 \text{ if } |u| < 1, \\ 0 \quad\quad \text{otherwise.}\end{cases}$$

In this paper, we assigned $\mathcal{R}_i$ for each observation (5 minute), where L_{MAD} has been obtained as

(6) $$L_{\text{MAD}} = \text{mediun}|\hat{\varepsilon}_i|.$$

Using (3), the next observation has been obtained for the estimated trend line, where observing any inequalities, the host can be identified as overloaded.

3.2.2. VM selection policy

In this phase, the VM selection takes place where it is intended to select the VM which should be migrated to minimize overhead from overloaded host. Estimating the dynamic CPU utilization threshold, VM selection has been performed that offloads host for avoiding SLA violation and unwanted energy consumption caused due to overload. In this paper, we have examined three different selection policies; the Minimum Migration Time (MMT), Maximum Correlation (MC) and Random Selection (RS) policy. A brief discussion of the implemented VM selection policies are given as follows:

3.2.2.1. Minimum migration time policy

Once assessing the host's CPU utilization levels and identifying any probable or overloaded host, VMs selection algorithm performs offloading of that host node to avoid any probability of SLA violation. The developed MMT selection policy performs migration of only those VMs (v) which requires minimal migration time than the other. In this paper, the migration time has been estimated in terms of the resource, RAM being used by VM divided by the supplementary network bandwidth available for host j. Let V_j be a set of VMs connected with the host j. Therefore, the VM to be migrated is selected based on the following condition:

$$v \in V_j | \forall_a \in V_j,\ \frac{\mathrm{RAM}_u(v)}{\mathrm{NET}_j} \leq \frac{\mathrm{RAM}_u(a)}{\mathrm{NET}_j}, \tag{7}$$

where $\mathrm{RAM}_u(a)$ depicts the amount of RAM currently being used by VMa; and NET_j refers the bandwidth available for migration from the host j.

3.2.2.2. Maximum correlation policy

In case of MC policy [25, 26] it is assumed that the higher correlation between the CPU utilization by VMs connected to certain host signifies higher overloading probability. In MC policy, VMs having the highest correlation for the CPU utilization are needed to be migrated from the current host to assist energy conservation and SLA violation avoidance. We have used the concept of Multiple Correlation Coefficients (MCC) for estimating the intra-VM correlation and respective CPU utilization. The used MCC coefficients are in relation to the squared correlation between the real and the predicted values of the dependent variable. In fact, it can be interpreted as the fraction of variance of the dependent variable elucidated by the associated independent variables.

Let $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n$ be the CPU utilization of n VMs connected to a host and $\mathcal{Y}$ be the specific VM to be migrated, where $n-1$ are the random independent variables. Here $\mathcal{Y}$ being the VM to be migrated, be the independent variable. In our proposed MC policy, we have calculated the correlation between $\mathcal{Y}$ and $n-1$. The obtained augmented matrix $(n-1) \times n$ encompasses the observed instances or the values of $n-1$, indicated as $\mathcal{X}$. Similarly, the observation vector or mapped vector $(n-1) \times 1$ of the dependent variable $\mathcal{Y}$ be y. Thus, the overall observation vector or the mapping vectors for the VM's CPU utilization can be given as follows:

$$\mathcal{X} = \begin{bmatrix} 1 & \mathcal{X}_{1,1} & \cdots & \mathcal{X}_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathcal{X}_{n-1,1} & \cdots & x_{n-1,n-1} \end{bmatrix},\ \mathcal{Y} = \begin{bmatrix} \mathcal{Y}_1 \\ \vdots \\ \mathcal{Y}_{n-1} \end{bmatrix}. \tag{8}$$

Observing (8), it can be found that the first column of $\mathcal{X}$ is all 1 (for all instances), therefore it can be considered as an augmented matrix. The predicted values of the VM's CPU utilization or $\mathcal{Y}$ can be presented as $\hat{\mathcal{Y}}$, which can be obtained by $\widehat{\mathcal{Y}} = \mathcal{X}b$, where $b = (\mathcal{X}^{\mathrm{T}}\mathcal{X})^{-1}\mathcal{X}^{\mathrm{T}}\mathcal{Y}$. Obtaining the predicted values, MCC coefficients, the final correlation $R^2{}_{Y,1,\dots,n-1}{}'$ has been obtained as

$$\mathcal{R}^2{}_{\mathcal{Y},\mathcal{X}_1,\dots,\mathcal{X}_{n-1}} = \frac{\sum_{i=1}^{n}(\mathcal{Y}_i - m_{\mathcal{Y}})^2(\hat{\mathcal{Y}}_i - m_{\hat{\mathcal{Y}}})^2}{\sum_{i=1}^{n}(\mathcal{Y}_i - m_{\mathcal{Y}})^2 \sum_{\mathrm{i}=1}^{n}(\hat{\mathcal{Y}}_i - m_{\hat{\mathcal{Y}}})^{2}}{}', \tag{9}$$

where $m_{\mathcal{Y}}$ and $m_{\hat{\mathcal{Y}}}$ give the observation means of $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ respectively. In MC based VM selection, the MCCs for all mapped instances $\mathcal{X}_i$ have been obtained as $\mathcal{R}^2{}_{\mathcal{X}_{\mathrm{i}},\mathcal{X}_1,\dots,\mathcal{X}_{\mathrm{i}-1},\mathcal{X}_{\mathrm{i}+1},\dots,\mathcal{X}_n}$. Finally, based on correlation value, (10) has been used to select a specific VM to be migrated:

$$v \in \mathcal{VM}_j | \forall_a \in \mathcal{V}_j, \mathcal{R}^2{}_{\mathcal{X}_{vm},\mathcal{X}_1,\dots,\mathcal{X}_{vm-1},\mathcal{X}_{v+1}\mathcal{X}_n} \geq \mathcal{R}^2{}_{\mathcal{X}_v,\mathcal{X}_1,\dots,\mathcal{X}_{a-1},\mathcal{X}_{a+1},\dots,\mathcal{X}_n}. \tag{10}$$

In addition to the MMT and MC selection policy, we have also examined an algorithm called RS policy.

3.2.2.3. Random selection policy

In RS policy, a VM is randomly selected for migration from the host node as per a uniformly distributed discrete random variable $\mathcal{A} = \mathrm{U}(0|\mathcal{V}_j)$, whose values signify a set of VMs, $\mathcal{V}_j$ placed at j-th host. Since VM consolidation is a bin packing problem and therefore, an optimal approach for placement is of great significance to ensure minimal downtime, energy consumption and probable SLA violation. The following section discusses the proposed evolutionary computing based VM placement approach.

3.2.3. VM placement policy

VM placement can be stated to be a problem of bin packing that encompasses bins, items and prices as the three parameters, where bins represent the host nodes, VMs represent the items to be allocated, bin size refers the available resource on the host node; and the resource or CPU power consumed by a host is stated in terms of price. In bin packing, it is intended to accommodate as much as VMs that makes the overall scenario NP-hard. In order to deal with such non-convexity problem, certain heuristic approach or evolutionary computing scheme can be the potential solution. In this paper, we have proposed A-GA as VM placement policy. We have interfaced A-GA with CloudSim simulator comprising multiple host nodes, and VMs in the data center. In our proposed model, each host is equipped with one or multiple Processing Elements (PE). The executing VMs on the hosts have one or multiple running Cloudlets. In simulation model, the user requests have been stated in terms of Cloudlets, where the needed processing power for each Cloudlet has been defined in terms of MIPS. In the proposed placement policy, the scheduler considers all hosts, VMs and VM maps as input and generates mapping for nodes, where it divides overall MIPS into different components like hosts and VMs running in parallel. The functional discussion of the proposed A-GA scheme is given in coming sections.

3.2.3.1. A-GA based VM placement

As depicted in Fig. 1, the proposed A-GA algorithm processes VM scheduling based on the resource utilization information provided by local controller and the upper threshold value estimated by dynamic threshold estimation scheme. We have considered upper threshold so as to satisfy transient variations of resource demand by different VMs on a host node. The CPU utilization pattern or history of VMs has been considered for VMs placement onto destination host node. Considering a large scale cloud infrastructure or data center, the time efficient and effective placement scheduling is of great significance. In this research we have intended to reduce the number of VM migration and migration time, so as to enable energy efficient and QoS oriented consolidation mechanism. In addition, we have scheduled the system to enable maximum host shut down so as to conserve energy. In this paper, placing or allocating VM on certain host, our algorithm estimates the energy of the data center and accordingly performs further scheduling to minimize energy consumption.

At first, the proposed A-GA algorithm initializes a definite set of population where the individual host is a tree comprising global controller as its root, the hosts are the next level nodes and VMs are the child nodes Fig. 1. It calculates the total energy consumption and CPU utilization for each mapping in the deployed cloud center. Here, the VM mapping history also known as utilization pattern, allocated VMs and their resource utilization mapping, future mapping for VMs based dynamic utilization, hosts and its available resource availability etc. have been used as population. The precise discussion of the proposed system is given in coming section. From these chromosomes, our proposed A-GA algorithm initially selects two VM mappings with minimal energy values on which the initial genetic operators (crossover p_{c} and mutation probability p_{m}) are applied. Thus, the mapping obtained for VMs onto the host nodes is added to the overall population based on the fitness values. In our proposed A-GA based VM placement policy, p_{c} selects the host with the best CPU utilization based on the previous VMs mapping. Here p_{c} and p_{m} try to reduce host nodes by means of SWITCHING OFF or turning it into SLEEP MODE. Here, it should be noted that unlike conventional genetic approach (i.e., A-GA), we have applied adaptive genetic parameter selection, where these variables are updated dynamically after every iteration, till stopping criteria (100 generations) is obtained.

Consider, the host nodes in data centers be

$$\mathrm{PM} = \{\mathcal{pm}_1, \mathcal{pm}_2, \mathcal{pm}_3, \cdots, \mathcal{pm}_m\},$$

and $\mathcal{pm}_i$ be ith host node, where $1 \le i \le m$. Similarly, VMs in the network be

$$\mathrm{VM}_i = \{\mathcal{vm}_1, \mathcal{vm}_2, \mathcal{vm}_3, \cdots, \mathcal{vm}_{n,i}\},$$

which are connected to the i-th host. Consider $\mathcal{vm}_{j,i}$ be the i-th VM on j-th PM. The variable $x_{i,j}$ ignifies whether i-th VM is placed on host j or not. Let $\mathcal{P}_{r,i}$ be the resource capacity r (CPU utilization) on j-th host node. The resource needed by i-th VM is $v_{r,j}$. Thus, overall load on j-th host node would be the sum of all resource needed by all VMs running over it. Consider, $\mathcal{T}$ be the duration of past observations, thus the sub-intervals can be obtained by dividing $\mathcal{T}$ into $q - 1$ sub intervals such that $\mathcal{T} = \left[(t_2 - t_1)(t_3 - t_2)\cdots\left(t_q - t_{q-1}\right)\right]$. The slot $t_k - t_{k-1}$ is the time period k. In such manner, for period k, we have estimated the CPU utilization at a host $\left(\mathrm{CPU}_{i,\mathrm{Util}}(k)\right)$ using following equation:

(11) $$\mathrm{CPU}_{i,\mathrm{Util}}(k) = \sum_{j=1}^{n} \mathcal{vm}_{\mathrm{CPU},j} / \mathcal{pm}_{\mathrm{CPU},j},$$

where, k represents the duration for which the CPU utilization has to be retrieved. Finally, the average CPU utilization at a host node has been obtained as:

(12) $$\mathcal{pm}_{i,\mathrm{AvgUtil}} = \sum_{t-t_k}^{t_k-n} \mathcal{pm}_{i,\mathrm{Util}}\,(t)/(q-1),$$

where $q - 1$ represents the total number of sub intervals in $\mathcal{T}$ time.

Consider $\mathcal{pm}_i$ represents the power of j-th host node during t_k. Thus, the power utilization can be obtained in terms of CPU utilization at the host node. Consider, $\mathcal{pm}_i\mathcal{E}^{(k)}$ be the power or energy consumption of the j-th host node in between the last time interval and the current time, then it can be obtained as

(13) $$\mathcal{pm}_i\mathcal{E}^{(k)} = \mathcal{pm}_{iw}(k-1) + (\mathcal{pm}_{iw}(k-1) + (\mathcal{pm}_{iw}(k))(t_k - t_k - 1).$$

The energy consumption for the j-th host, $\mathcal{E}(pm_j)$ can be estimated at certain host pm_j having CPU usage as $\mathrm{CPU}_{i,\mathrm{Util}}(k)$

(14) $$\mathcal{E}(pm_j) = \mathcal{K}_j \,.\, e_j^{\max} + (1 - k_j) \,.\, e_j^{\max} \,.\, \mathrm{CPU}_{i,\mathrm{Util}}(k),$$

where $\mathcal{K}_j$ states the part of energy consumed when the host pm_j is in idle state; $e_j^{\max}$ states for the energy consumption of host pm_j when it being used 100%. The variable $\mathrm{CPU}_{i,\mathrm{Util}}(k)$ represents the CPU utilization by host pm_j. We have used this approach to estimate the energy consumption at certain host so as to perform placement scheduling. Similarly, the energy consumption for all hosts $\mathcal{D}_{\mathcal{E}}(k)$ can be obtained for a period using following equation:

(15) $$\mathcal{D}_{\mathcal{E}}(k) = \textstyle\sum_{i=1}^{m} pm_i \mathcal{E}^{(k)}.$$

In this paper, the prime objective of the proposed A-GA scheme for VM placement is to retrieve the set of mapping from VM set to the host set PM while ensuring minimal energy consumption $\mathcal{D}_{\mathcal{E}}(k)$, provided:

(16) $$\forall_i \textstyle\sum_{j=1}^{m} x_{ij-1},$$

(17) $$\forall_j \textstyle\sum_{i=1}^{n} vm_{\mathrm{CPU},i} \mathcal{X}_{ij} \le pm_{\mathrm{CPU},j}.$$

The implementation of our proposed A-GA based consolidation scheme is as follows:

Step 1. Create VMs with random workloads and allocate them randomly on PMs.

Step 2. Perform each iteration at a defined scheduling interval (say 5 min) to estimate CPU Utilization and dynamic thresholding based resource prediction until Simulation Limit expires

Odd iterations: Calculate upper threshold value for all PMs using IQR.

Even iterations: Calculate upper threshold value all PMs using LRR.

Step 3. Perform Initial mapping of VMs and PMs and add it as initialchromosome to population (called root).

Step 4. Estimate the list of overloaded PMs using Step 2.

Step 5. Estimate the list of VMs from each host that are not under migration.

Step 6. Perform VM selection for all overloaded or over-utilized PMs and obtain the list of all VMs that are ready for migration.

Step 7. Sort all VMs based on their CPU Utilization and allocate them tooverloaded PMs, while ensuring that these hosts should not get overloaded again.

Step 8. Obtain a minimalHost from the list of active hosts having minimal resource CPU utilization.

Step 9. Allocate all VMs from minimalHost to active hosts while ensuring that allocation would not cause overloading on that host node.

Step 10. Current mapping for VMs and PMs is retrieved as proposed partner chromosome and add it to the A-GA population.

Step 11. Obtain the first child by performing initial crossover between the root (Step 3) and proposed partner (Step 10) on the basis of candidate fitness value:

$$\text{Candidate fitness value} = \frac{\text{MIPS Utilized by all VMs on a PM}}{\text{PM'sMIPS}}.$$

Step 12. Mutate the first child chromosome and add to population by allocating some VMs randomly to PMs.

Step 13. Calculate the Roulette wheel (Rwheel) value using probability fitness:

$$\text{Probability fitness} = \frac{\text{CandidateFitnesValue(Candidate, root)}}{\text{Cumulative fitness value}},$$

$$\text{Rwheel} = \frac{\text{Probability Fintness Value}}{\text{Number of PMs}}.$$

Step 14. Select parent nodes (Father and Mother) using Rwheel probability value to perform crossover.

Step 15. Mutate the chromosome obtained from Step 14 and add it to the population.

Step 16. Perform Steps 13-15 till stopping criteria is met.

(We have defined 100 as the total number of population (generations), i.e., stopping criteria).

Step 17. From population, obtain the Best VM/PM map having higher candidate fitness value for placement.

Step 18. Allocate VMs to PMs using Step 17.

Step 19. Supply this map to Step 2 for next scheduling purpose.

In our approach, after every iteration the value of the Roulette wheel probability, which is used for crossover, gets changed and thus it exhibits adaptive nature of GA. Due to this reason, our proposed scheme has been named as Adaptive Genetic Algorithm (A-GA). In this paper, in addition to the A-GA based VM placement policy, other heuristic approaches,such as ACO and BFD have been employed for placement policy and respective performance comparison has been done.

4. Experimental setup

In this paper, we have examined the performance of the proposed system using real time cloud workload traces retrieved from CoMon data project, which is a part of Cloud monitoring infrastructure of PlanetLab [27]. To evaluate the robustness of the proposed consolidation scheme a large scale cloud infrastructure containing 1000s of VMs and host nodes (PMs) has been considered. The benchmark cloud image or workload traces encompass the CPU utilization of 1000+ VMs connected with the servers located at 100s of different places. The workload traces have been obtained during 10 randomly selected days in March and April 2011 and CPU utilization has been measured at the interval of 5 minutes. Java-Eclipse has been used for programming and CloudSim platform has been used for simulation. We have used to distinct servers with different configurations.At first, the frequency of the servers has been mapped onto MIPS ratings where the individual server is mapped with 1860 and 2660 MIPS in HP ProLiant ML110 G4 and HP ProLiant ML110 G5, respectively. Each server has been assigned 1 GB per 1 s network bandwidth.

5. Results and discussion

The performance evaluation of the proposed research model has been done with a different combination of overload detection schemes, VM selection policies and placement policies. The performance of proposed A-GA based placement policy for consolidation has been compared with the other heuristics like ACO and BFD. The performance has been compared with different approaches like conventional IQR, LR, MAD, THR algorithm based CPU utilization estimation scheme and MMT, MC and RS based VM selection. Here, it should be noted that the other existing approaches (IQR, LR, MAD, THR and LRR) have been employed with BFD based VM placement algorithm only Table 1.

Table 1. Implementation and simulation scenarios

CPU utilization threshold	VM selection			VM placement
IQR	MMT	MC	RS	BFD
LR	MMT	MC	RS	BFD
MAD	MMT	MC	RS	BFD
THR	MMT	MC	RS	BFD
LRR	MMT	MC	RS	BFD
Combined IQR and LRR	MMT	MC	RS	ACO
	MMT	MC	RS	A-GA

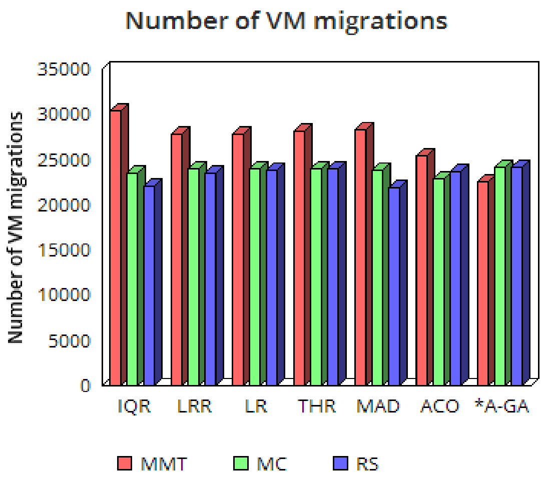

Fig. 2. Number of VM migrations

Considering better efficiency of IQR and LRR, we have used these algorithms with our proposed A-GA based consolidation scheme. We have examined the performance of different techniques under three distinct VM selection policies, MMT, MC and RS. Based on different threshold estimation (for overload detection), VM selection and placement policies, the simulation has been done and numerous performance parameters such as energy efficiency, SLA violation, SLA time per active host, number of host shut down etc., have been assessed. The different simulation scenarios and respective results outcome are given as follows:

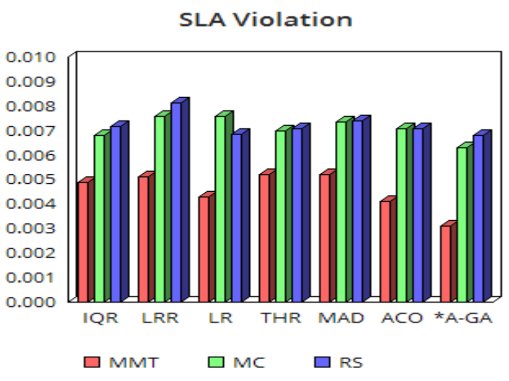

Fig. 3. SLAV (%)

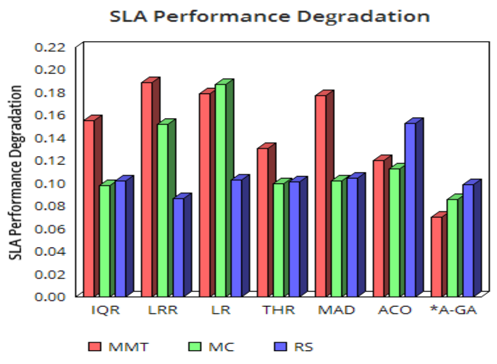

Fig. 4. SLA performance degradation (%)

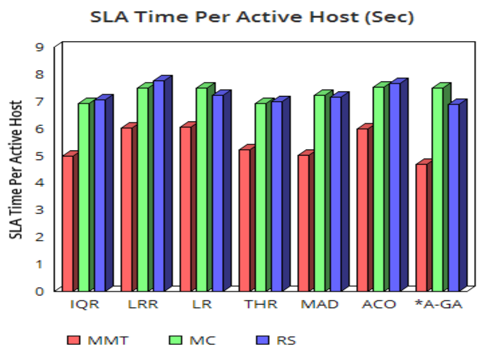

Fig. 5. SLA time per active host (s)

To perform better analysis, we have examined different algorithms with three different VM selection and placement policies. As depicted in Fig. 2, the proposed A-GA based consolidation exhibits minimal migration thus enabling minimal downtime probability. Fig. 3 represents the SLAV, where the proposed evolutionary computing based proposed system has exhibited minimal SLA violation with MMT selection policy. Fig. 4 depicts the SLA performance degradation, where it can be observed that the proposed A-GA based VM placement strategy with MMT VM selection policy and LRR based overload detection and resource prediction can enable minimum performance degradation. Similarly, Fig. 5 affirms better performance of the proposed system in terms of SLA per active host. Fig. 6 states that Combined IQR and LRR+MMT+A-GA based consolidation scheme can perform better as compared to other possible consolidation scenarios. In addition to the QoS and reliability, we have examined the energy efficiency of the proposed VM consolidation technique. The results obtained in Fig. 7 represents that the proposed evolutionary computing based VM placement scheme ensures minimal power consumption. Observing the results retrieved, it can be found that the minimal migration, maximum host shut down enables this significant energy conservation.

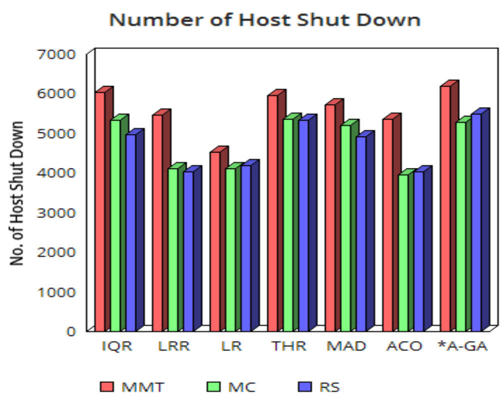

Fig. 6. Number of hosts shut down

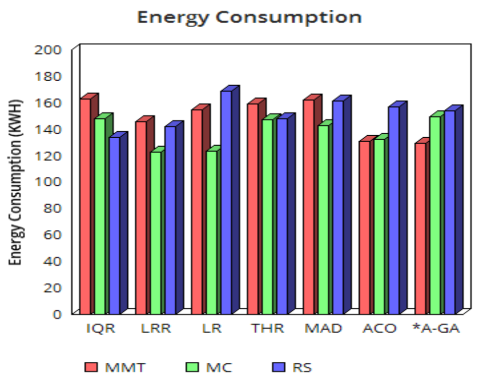

Fig. 7. Energy consumption (kW.h)

6. Conclusion

In this paper, a highly robust and efficient A-GA based on VM consolidation scheme has been developed, where different issues of the large scale cloud infrastructures like dynamism in resource utilization, number of VM migration, downtime reduction, SLA violation, and energy efficiency, etc. have been considered for optimization. The implementation of a combined IQR and LRR for dynamic threshold estimation has performed better for resource utilization and thresholding than other algorithms such as LR, conventional LRR, MAD, THR, etc. Furthermore, MMT based VM selection policy with proposed A-GA based VM placement or allocation has performed better than MC and RS scheme for energy efficiency and QoS in terms of minimal downtime and SLA violation. A-GA has exhibited better the other heuristics such as ant colony optimization and best fit decreasing based VM consolidation. The proposed system depicts minimum VM migration and SLA violation, low SLA active per host, and maximum energy efficiency as compared to other system.Hence, the proposed system can be used for large scale cloud resource management; energy efficient and QoS oriented cloud service provisioning.

References

1. F r a s e r, C. K., et al. Live Migration of Virtual Machines. – In: Proc. of 2nd USENIX Symposium on Networked Systems Design and Implementation, Berkeley, CA, 2005, pp. 273-286.
2. V o g e l s, W. Beyond Server Consolidation. – ACM Queue, 2008, No 1, pp. 20-26.
3. F e l l e r, E., C. M o r i n et al. A Case for Fully Decentralized Dynamic VM Consolidation in Clouds. – In: Proc. of 4th IEEE International Conference, Cloud Computing Technology and Science, Taipei, Taiwan, 2012, pp. 26-33.
4. M u r t a z a e v, S. O. Sercon: Server Consolidation Algorithm Using Live Migration of Virtual Machines for Green Computing. – IETE Technical Review, Vol. **28**, 2011, No 3, pp. 212-231.
5. M a r z o l l a, M., O. B a b a o g l u et al. Server Consolidation in Clouds through Gossiping. – In: Proc. of 12th IEEE International Symposium, World of Wireless, Mobile and Multimedia Networks, Lucca, Italy, 2011, pp. 1-6.
6. B e l o g l a z o v, J. A., et al. Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. – Grid Computing and e-Science, Future Generation Computer Systems, Vol. **28**, 2012, pp. 755-768.
7. B e l o g l a z o v, R. B. Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers. – Concurrency and Computation: Practice and Experience, Vol. **24**, 2012, No 13, pp. 1397-1420.
8. F a r a h n a k i a n, F., P. L i l j e b e r g et al. Linear regression Based CPU Usage Prediction Algorithm for Live Migration of Virtual Machines in Data Centers. – In: Proc. of 39th Euromicro Conference of Software Engineering and Advanced Applications, Santander, Spain, 2013, pp. 357-364.
9. F a r a h n a k i a n, F., T. P a h i k k a l a et al. Energy Aware Consolidation Algorithm Based on K-Nearest Neighbor Regression for Cloud Data Centers. – In: Proc. of 6th IEEE/ACM International Conference on Utility and Cloud Computing, Dresden, Germany, 2013.
10. W o o d, T., P. S h e n o y et al. Sandpiper: Black-Box and Gray-Box Resource Management for Virtual Machines. – Computer Networks, Vol. **53**, 2009, pp. 2923-2938.
11. A j i r o, Y., A. T a n a k a. Improving Packing Algorithms for Server Consolidation. – In: Proc. of International Conference for the Computer Measurement Group, San Diego, California, USA, 2007, pp. 399-407.
12. W a n g, M., X. M e n g et al. Consolidating Virtual Machines with Dynamic Bandwidth Demand in Data Centers. – In: Proc. of 30th IEEE International Conference on Computer Communications, Shanghai, China, 2011, pp. 71-75.
13. H a r m a n, M., K. L a k h o t i a et al. Cloud Engineering is Search Based Software Engineering Too. – Journal of Systems and Software, Vol. **86**, 2013, No 9, pp. 2225-2241.
14. D o r i g o, M., G. D i C a r o et al. Ant Algorithms for Discrete Optimization. – Artificial Life, Vol. **5**, 1999, No 2, pp. 137-172.

15. D o r i g o, M., L. G a m b a r d e l l a. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. – IEEE Transactions on Evolutionary Computation, Vol. **1**, 1997, No 1, pp. 53-66.
16. B a r b a g a l l o, D., E. D i N i t t o et al. A Bio-Inspired Algorithm for Energy Optimization in a Self-Organizing Data Center. – Self-Organizing Architectures, Springer, 2010, pp. 127-151.
17. C h e n , H., L. X i o n g et al. Cloud Task Scheduling Simulation via Improved Ant Colony Optimization Algorithm. – Journal of Convergence Information Technology, 2013.
18. D o n g, Y. S., G. C. X u et al. A Distributed Parallel Genetic Algorithm of Placement Strategy for Virtual Machines Deployment on Cloud Platform. – The Scientific World Journal, 2014, pp. 1-12.
19. F e l l e r, E. E. et al. Energy-Aware Distributed Ant Colony Based Virtual Machine Consolidation in IaaS Clouds Bibliographic Study. – Informatics Mathematics (INRIA), 2012, pp. 1-13.
20. F e r d a u s, M. H., M. M u r s h e d et al. Virtual Machine Consolidation in Cloud Data Centers Using ACO Metaheuristic. – In: Proc. of 20th International Conference Euro-Par 2014 Parallel Processing, Porto, Portugal, 2014, pp. 306-317.
21. Z h o n g, H., K. T a o et al. An Approach to Optimized Resource Scheduling Algorithm for Open-Source Cloud Systems. – In: Proc. of China Grid Conference (China Grid), 2010, Fifth Annual, Guangzhou, China, 2010, pp. 124-129.
22. M a d h u s u d h a n, B., K. C. S e k a r a n. A Genetic Algorithm Approach for Virtual Machine Placement in Cloud. – In: Proc. of International Conference on Emerging Research in Computing, Information, Communication and Applications, 2013, pp. 115-122.
23. T a n g, M., S. P a n. A Hybrid Genetic Algorithm for the Energy-Efficient Virtual Machine Placement Problem in Data Centers. – Neural Processing Letters, Vol. **41**, 2015, No 2, pp. 211-221.
24. C l e v e l a n d, W. S. Robust Locally Weighted Regression and Smoothing Scatterplots. – Journal of the American Statistical Association, Vol. **74**, 1979, No 368, pp. 829-836.
25. V e r m a, G. D. et al. Server Workload Analysis for Power Minimization Using Consolidation. – In: Proc. of 2009 USENIX Annual Technical Conference, San Diego, China, 2009, pp. 28-42.
26. A b d i, H. Multiple Correlation Coefficient. – N. J. Salkind, Ed. Sage, Thousand Oaks, CA, USA, 2007.
27. P a r k, K. S., V. S. P a i. CoMon: A Mostly-Scalable Monitoring System for Planet-Lab. – ACM SIGOPS Operating Systems Review, Vol. **40**, 2006, No 1, pp. 65-74.
28. T h e j a, P. R., S. K. K. B a b u. An Evolutionary Computing Based Energy Efficient VM Consolidation Scheme for Optimal Resource Utilization and QoS Assurance. – Indian Journal of Science and Technology, 77179, Vol. **8**, 2015, No 26, pp. 1-11.
29. T h e j a, P. R., S. K. K. B a b u. An Adaptive Genetic Algorithm Based Robust QoS Oriented Green Computing Scheme for VM Consolidation in Large Scale Cloud Infrastructures. – Indian Journal of Science and Technology, 79175, Vol. **8**, 2015, No 27, pp. 1-13.