International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 02, 2020
ISSN: 1475-7192

AN ENERGY EFFICIENT AND SELF ADAPTIVE RESOURCE ALLOCATION FRAMEWORK USING MODIFIED CLONAL SELECTION ALGORITHM FOR CLOUD BASED SOFTWARE SERVICES

[1]Dr Balamurugan E, [2]Md. Shahidul Hasan, [3]Mohammad Shawkat Akbar Almamun, [4]Sangeetha K

Abstract: *Cloud computing is a prominent model for computation which lays greater impact on IT industries and the way in which software applications are developed and deployed. The workload in cloud computing changes dynamically and thereby introduces many challenges in on-demand resource provisioning and allocation. Resource allocation techniques are not widely available. Those that exist also have high energy consumption as well as the ineffectiveness of allocation. Frequent virtual machine switches in Virtual Machine allocation leads to a tradeoff between QoS and consumption of energy. Hence there is a need to find a solution which provides a quality of service and low energy consumption for use in cloud services. The proposed work suggests self-adaptive framework for allocation of resources composed of feedback loops to meet the QoS requirements as well as less energy consumption in cloud computing services. The self-adaptive resource allocation framework comprises of three stages namely QoS prediction model, improved cuckoo search algorithm (ICSA) - based runtime decision algorithm and Energy Efficient Model (EEM).*

In the First stage, QoS prediction model works over historical data of a system that aids in improving the accuracy rate of QoS prediction. Second an Energy Efficient Model based on Modified Clonal Selection Algorithm (MCSA) has been suggested for reducing the energy consumption. Third stage is a runtime decision-making algorithm which works according to the ICSA helps to find proper operation for allocating resource in an online real context. Experiments are conducted and the results convey that the proposed work could decrease the number of times hosts are switched on/off. This technique when compared to existing methods, helps to save power, provides cost effective and improves the QoS for cloud computing environments.

[1] University of Africa, Toru-Orua, Nigeria,
[2] Research Scholar, Texila American University, Guyana
[3] Research Scholar, Texila American University, Guyana
[4] University of Africa, Toru-Orua, Nigeria,

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

Keywords: *IT industries, Virtual Machine, Modified Clonal Selection Algorithm (MCSA)*

I. INTRODUCTION

In the emerging era cloud computing services play a leading role of prominent technology to deliver the solutions and services through Internet. It uses a collection of virtualized resources for computation [1].Customers using the cloud services began to rise exponentially. To meet their need many services providers in cloud such as Amazon, Microsoft, Google started increasing the data centers significantly. This created the issues such as high energy consumptions and large emission of CO2. Recent surveys indicate that utilization of energy by the data centers is as equal to the energy usage value of twenty-five houses. Electricity consumption of data centers is raised by 56% between the years 2005 and 2010.If this progress continues, may be observed that annual energy will outpace the equipment price [2].Also on further assessment ,Information and communications field accounts to 2% carbon radiations globally and datacenters contribute 14%. Because of the monetary and natural effect of energy utilization, there is an expanded enthusiasm for the advancement in energy utilization in the services associated with cloud systems. This helps in reducing operational cost of a data center and protects the environment from CO2 emissions [3].

Infrastructures that support cloud computing for accessing and deploying service oriented applications are developed for the user. These services are promoted through data centers. They also offer high speed and huge volume computational and storage servers to contribute to the demand of computation and storage. These have higher power consumption and hence they need air conditioning facilities to dissipate its heat [5].The energy conservation is almost in proportion to how we utilize a resource With respect to all the resources, data centers consumes more electricity .Cloud computing offers services in three modes namely SaaS (Software as a Service), IaaS (Infrastructure as a Service) in conjunction with RaaS (Resource as a Service) , PaaS (Platform as a Service).It promotes efficient resource management and resource mobility[6].

It deals with many issues of how to manage resources, and migration of resources to mobile users [6]. A notable issue that disrupts the computing power and battery life has been overcome.

Cloud technology competes to next generation by designing data centers as a interconnection of virtual services (user-interface, database, processing logic, etc.).This helps the users from across the world to use and deploy their applications at a marginal cost based on the QoS (Quality of Service) [7].Developers of multiple domain can avail this cloud service with no extra hardware cost nor manpower for its operation. It also adds more advantages to the IT companies to concentrate on innovative ideas and business values without thinking of the infrastructure and hardware needs. Resource management is a leading task for cloud services in areas of operating systems, managing the data centers and grid computing of. It effectively allocates the resources among users and applications of the cloud economically [8].In IaaS, resource management is more demanding and offers more advantages. It is more profitable as users don’t need to have a hardware or software setup and they can access its services from any part of the world with no usage ceiling.

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

Considering the spectral efficiency (SE), Energy efficiency (EE) also plays an important factor for designing next generation network. It also takes into account the energy and power consumption [9].Load, cost and speed are the criteria for allocating resources in virtual environment. To meet the demand of the application skewness concept is used in the virtualization field.VM allocation is secure, especially for multi-users using the same infrastructure. There are two strategies, namely VM placement and VM selection in the virtual resource allocation. The VM placement arranges the available VMs in descending order according to CPU utilization. Then for every host it distributes a host which will ensure power consumption in the least increasing order. There are two steps in the VM selection in which it chooses the VMs needed to migrate in the first step. Recently the self-adaptive resources allocation became a challenging task where by cloud service updates its resources based on its current need [10].Conservative self adaptive methods are based on rules ,where by every service is associated with a rule for making optimal decisions. But those methods are difficult to implement and consumes more overhead in terms of implementation.

Heuristical algorithms are in place which provides a particular resource allocating technique based on QoS predictions. But large volume of data is needed for building the prediction system. Generally these historical data are insufficient and has lesser variances which cannot handle different workloads and resources. So this prediction model of QoS is not effective to support accuracy and resource allocation. Another technique is the control theory that uses feed-back controller to achieve QoS. It works by adjusting the system behavior with respect to the output dynamically. This is performed by changing the input parameters which has the influence on the virtual machine type and their count. These techniques help to specify how the control elements interact with each other and they maintain the system attributes during runtime. The iteration increases to select an associated resource technique that is a overhead. Hence virtual machines are not used in this method.

This research work provides a self-adaptive resource allocation framework composed of feedback loops to meet the QoS requirements along with less energy utilization to be used for cloud services. This self-adaptive resource allocation framework comprises of three stages namely QoS prediction model, improved cuckoo search algorithm (ICSA) - based runtime decision algorithm and Energy Efficient Model (EEM) to be used in the cloud services.

The remaining work is classified in a way that section 2 elaborates the existing resource allocation techniques, section 3 explains about proposed methodology, section 4 illustrates the results and discussion ,section 5 tells about the conclusions and future work .

II. LITERATURE REVIEW

Goudarzi et al [12] developed a Service Level Agreement (SLA) through which resource allocation issue could be solved for multitier applications used in cloud software services. The proposal works through an algorithm using force-directed search, given limit on the net profit. The three parameters such as memory, computation and communication are considered for optimization. Simulations are done for improving the efficiency of the algorithm that uses heuristic technique.

Wang et al [13] suggested auction models that drive various resource allocation techniques. It works through auction based technique for provisioning resources used by the cloud environment. Initially a framework is developed for allocating resources in the cloud environment and then its problems are described while designing auction based model for resource allocation.

Ram Mohan et al [14] developed an Interference Aware Resource Allocation (IARA) technique. IARA technique schedules the resources for reaching the optimistic level in the cloud computing paradigm. The proposed method is developed using various hardware to localize it and to make it work in any cloud environment where there is a resource clinch. The proposed IARA which uses scheduling process performs well in allocating resources and in utilizing the system resources. Experiments are conducted using simulation to verify the effectiveness of resource used, utilization of energy and evaluation time.

Xiong et al [15] proposed an algorithm to allocate virtual machine through an efficient multi resource allocation algorithm and through particle swarm optimization (PSO) process. The algorithmic work uses fitness function as a cumulative Euclidean distance which helps in finding the optimal point among energy consumption and resource consumption. It does not get into local optimal as like traditional heuristic algorithms. Also energy conservation and optimization of resources are considerable in this method.

Sharma et al [16] developed an algorithm using the combination of Dynamic Voltage Frequency Scaling (DVFS) and Genetic algorithm (GA) to allocate resources optimistically which is more effective in conserving energy. Performance of this algorithm is checked with DVFS algorithm. Experiments are conducted and results indicate that energy consumption is 22.4% less with a particular workload and there is 0%SLA violation.

Wang et al [17] developed a particle swarm optimization (PSO) algorithm used for allocating resources consumes less energy. The new algorithm has a new coding technique which changes the operators and parameters in PSO by using local fitness first technique. By applying this algorithm a new replacement method for virtual machines which conserves less energy is developed. . Experiments are conducted and results convey the proposed work can decrease the energy utilization by 13-23%.

Liu et al [18] proposed a technique namely Ant Colony Optimization (ACO) which sorts out the Virtual Machine Placement (VMP) issue called ACO-VMP. This concentrates on consuming less physical servers and using resources efficiently. Initially physical servers and VMs count remain same.ACO technique reduces the servers count gradually. ACO-VMP performance is evaluated considering the VMs to be 600 to solve VMP issue. Experiments are done and the results say that this algorithm outperform FFD algorithm by reducing the physical servers to a larger extent when VMS exist in a large number.

Joseph et al [19] designed a new process using Family Gene technique for allocating virtual machines. The proposed work functions by taking a list of host and mapping it with a list of VM .FGA module splits the whole process with different families which runs in a module through paralleled mode. Results are good that convey the family genetic algorithm could be used in physical data centers. The consumed energy level is low so that it could be applicable for green data centers. This algorithm is tested in cloud environment which could also be used by any

Tang et al [20] introduced a genetic algorithm to conserve energy used in the system and network and addresses the virtual machine placement issue. This research work increases the performance and efficiency by using a hybrid approach .Experiments is conducted and the results convey that these hybrid algorithms are scalable and the performance is far better than the conventional genetic algorithm.

Marphatia et al [21] proposed an optimized version of the First Come First Served (FCFS) scheduling algorithm. It overcomes many problems in scheduling tasks in the cloud environment. This is done by grouping the tasks based on low execution, low cost and priority level is set according to it. Greedy technique is used to select the resources by applying the task constraints to it. Simulation toolkit is used to implement and test the proposed work. Furthermore, expect to make a module portraying the typical FCFS model in contrast with the proposed work for resources selection in the cloud computing field.

Srinivasa et al [22] designed a technique called Min-Max process for solving the issues that occur while allocating resources in the cloud environment. Furthermore utility maximization technique is used for solving the resource related issues. A new parameter utility factor is used where time and budget constraints are levied upon each consumer. Resources are provided for every task based on the utility factor.

Chen et al [23] designed a technique for allocating resources which is self adaptive in nature. It is a framework comprising feedback loops and for every loop QoS prediction and PSO-based runtime decision algorithm is applied. Existing prediction models finds the QoS value only once but this model uses iterations to improvise the QoS value. In this prediction workload, resources allotted and QoS value is used first. Later PSO-based runtime decision uses QoS value also to find the method for allocating resources. Loops get iterated until the fine tuning of resource allocation is achieved. This approach is assessed using RUBiS benchmark using the historical data. Results say that it has 15% more accuracy rate compared to the existing state. Efficiency of allocating resources is achieved by 5-6%.

From the above literature analysis, many concepts and algorithms are available for allocating resources in cloud environment. However the Energy conservation affects the operational overhead of the allocation. The various algorithms namely particle swarm optimization, Min-Max Game approach, Genetic algorithm and greedy algorithms are not effective for efficient resource allocation. Energy consumption depends on the workload of requests and the active state of systems.

III. PROPOSED METHODOLOGY

In this research work, an adaptive autonomic method for reconfiguring virtual machines has been developed to meet the increasing demand of the workload. This self-adaptive resource allocation framework composed of feedback loops as iterative QoS prediction model, improved cuckoo search algorithm (ICSA) - based runtime decision algorithm and Energy Efficient Model (EEM) and later a resource allocation method is created to meet the cloud services.

- In this approach QoS prediction model based on iterations has been developed which uses historical data can provide more accuracy.

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

- An Energy Efficient Model based on Modified Clonal Selection Algorithm (MCSA) is developed for reducing energy intake. .
- A runtime decision-making algorithm uses ICSA, finds a particular process for allocating resources in a real time context.
- It makes use of QoS prediction model, Energy Efficient Model and runtime decision making algorithm

that comprises feedback loops to be used in the proposed work to implement a resource allocation framework with feedback and using iterations.

The overall process of the proposed self-adaptive resource allocation framework is illustrated in figure.1.

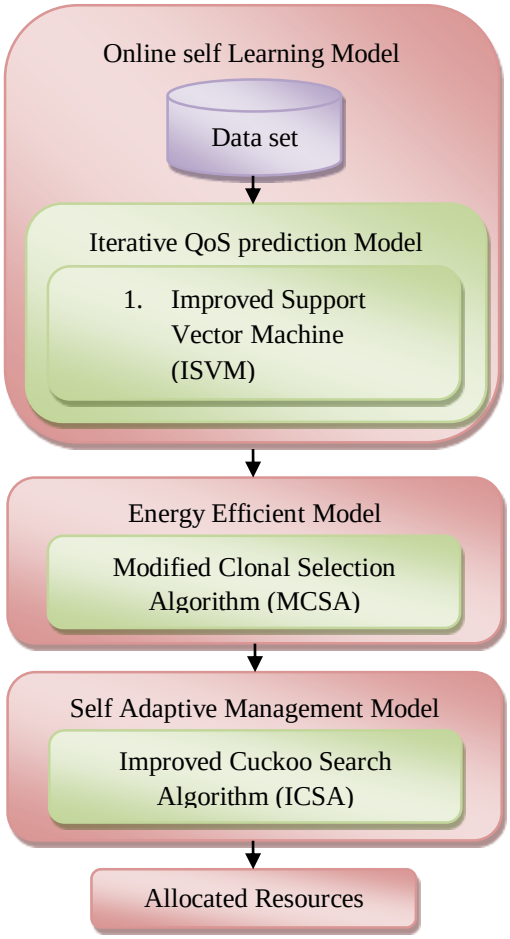

Figure .1. The overall process of the proposed self-adaptive resource allocation framework

3.1 System model

As the runtime environment undergoes a significant change, it has a major effect in the quality of the cloud services. Based on the initiating factors [23], changes in the environment could be classified as internal and external. External factors represent the workloads (L) and the proposed work tells about its changes. Internal factors tell about the allocated resources. Data center utilizes more energy which are due to the memory, CPU, power supply, cooling systems and mass storage devices. Energy consumption could be represented in an equation (1).

The external factors such as workloads (L) and their changes are specified in this research work along with the internal factors namely allocated resources (VM). Energy utilization of a data center is computed from many factors namely CPU utilization, memory utilization, power supply units, disk storage boxes and cooling systems. Energy consumption could be framed through equation (1).

$$Energy\ Consumption = \int Power\big(u(t)\big)dt \qquad (1)$$

Here u (t) represents the CPU usage. Power consumption is studied and is given with a linear representation with CPU usage. Henceforth CPU utilization is in direct proportion with power consumption as represented in equation (2).

$$Power(u) = q.P_{max} + (1 - q)P_{max}.u \qquad (2)$$

Here q represents the fraction of idle server's energy consumption, P_{max} *sp*ecifies the maximum power used by a server and u represents how much CPU is utilized.

Self-adaptive systems must have a equivalence among quality of services (QoS) and resources cost (Cost) during resource allocations in a cloud environment that depend on the defined targets. Fitness function is the normal process used to find the values given by a formula (3):

$$Fitness = r_1 * \frac{1}{QoS} + r_2 * Cost + r_3 * Energy \qquad (3)$$

Where r1, r2 and r3 are the parameters that define the weightage of QoS cost of the resource and the resource energy consumption. Besides these QoS value and the cost of resource is computed in the succeeding section.

When Fitness function value is less, it signifies a resource allocation technique which is devised depending on the real applications. Hence based on the workload, could predict the estimation value of every resource allocation technique and decisions could be made more effectively.

Based on real-time applications, when we choose a best resource allocation strategy, fitness function would yield a smaller value. Hence it could give a better prediction value for very resource allocation method with the workload and it could have better decision making capacity.

According to practical applications, a better resource allocation plan would result in a smaller value of fitness function. Therefore, can predict evaluation values of each possible resource allocation plans under the current workload, and make more effective decisions. Based on the equation (3), fitness value is decided by two factors. One is resource cost that is derived from the leased cost (CostL) and the other one is the discontinued cost (CostD) of virtual machines. It is given by the formula

$$Cost\ =\ CostL\ +\ CostD \qquad (4)$$

In the formula, CostL specifies the accumulated cost of all virtual machines and CostD specifies the penalty cost for shutting down all virtual machines called as discontinued cost. Frequent changes will lead to incur more computation and resource cost for the system. The discontinued cost aims to reduce the costs that are unnecessary while shutting down the virtual machines and helps to have a stable state of the software.

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

The second parameter QoS is found using the Service Level Agreements (SLA) contract and given by the following formula (5)

$$Q_{actual} = SLA(RT, DH, \dots,), \quad (5)$$

Where RT and DH represents the time taken for the response and data throughput. Numerous factors has a direct influence in QoS values. Among it are the response time and the throughput that is based on the software service type and IO and CPU intensive. Indeed there are other variables that influence QoS value, factors such as response time and data throughput should be concentrated that is based on the software services type, IO or CPU-intensive. In particular, proposed QoS model provides more expertise in allocating resources. This model used the workload, resources, resource adjustment action value, present QoS values as inputs and the resultant QoS value is obtained or predicted as output after adjusting the resources.

3.2. Adaptive resource allocation method based on feedback loop

The proposed work uses a resource allocation framework which is self-adaptive in nature works by continuous iterations and feedback loops.Fig.3 shows the systematic diagram of this framework. Instead of using historical data [23] which is inadequate, this algorithm uses feedback integrated with the trained data. The main motive is to use the feedback control technique along with the machine learning technique. It helps to produce a best value for QoS. The accuracy rate of machine learning processes could be improved. This framework comprises three modules namely Online Self learning module Energy Efficient module, Self-adaptive Management module. Cloud Resource module helps to oversee the resources and its statuses through its API .with this resources could be adjusted based on the need. Online Self-learning module makes use of legacy data to develop an iterative QoS prediction model through training by two methods namely Improved SVM (ISVM), Deep Neural Network (DNN). Self-adaptive Management module uses resource adjustment technique developed on iterative QoS prediction model used in automatic decision-making.

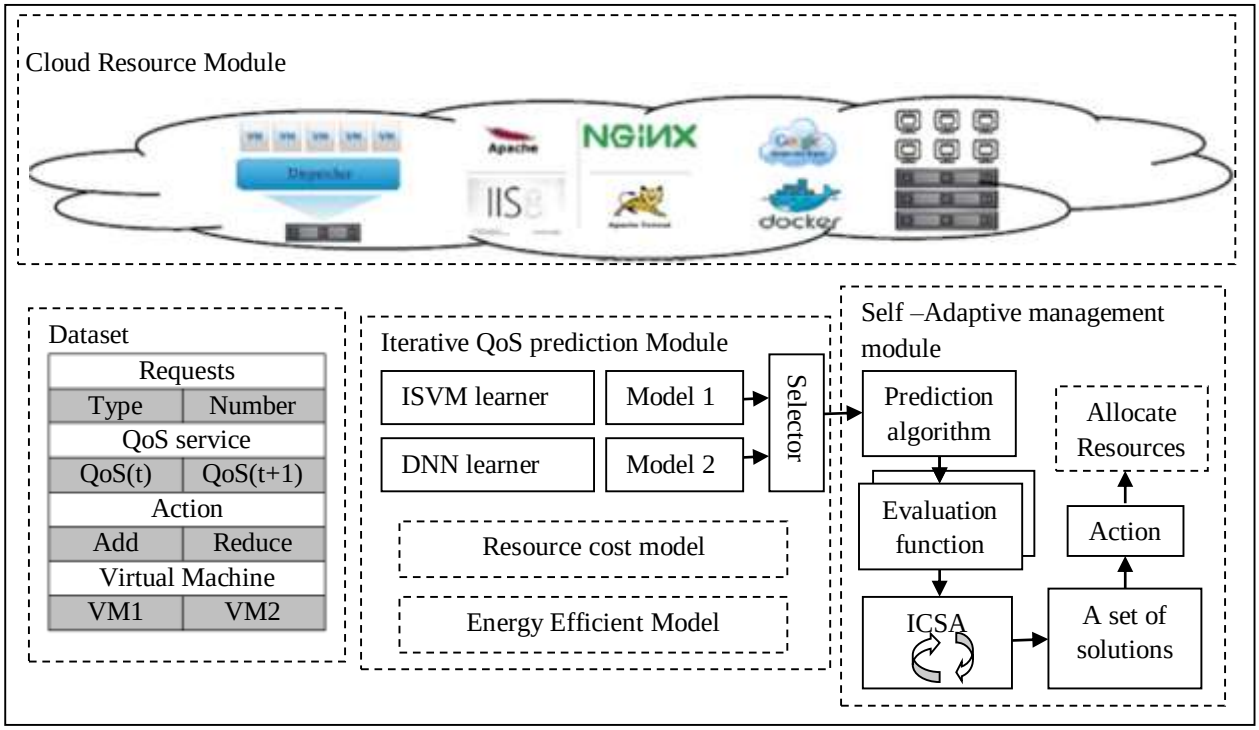

Figure .2. Detail process - proposed self-adaptive resource allocation technique

A. Iterative QoS prediction model

In the current section Online Self-learning module is discussed which tells about the training data used in prediction of the QoS model. This model used the workload, resources, resource adjustment action value and present QoS values as inputs and resultant QoS value is obtained or predicted as output after adjusting the resources. It is given by the formula (6):

$$QoSt + 1 = F\,(L, VMs, QoSt\,, Action), \qquad (6)$$

Here inputs L represents the current running workload, VMs represent the no of allocated virtual machines , QoS represents the service quality. Action represents adjustments made by virtual machines as such as adding or removal of virtual machines. The output QoSt+1 represent the service quality after Action. The historical data is used as a dataset for training iterative QoS model, given by Table 1.Workload is given by (xi,0 xi,1 . . . xi,m), here xi,0 refers to the amount of workload and xi,j($1 \leq j \leq m$) refers to the proportion of various tasks present in the workload. Resources allocated is indicated by (xi,m+1 xi,m+2 . . . xi,m+n), here xi,m+p denotes the count of virtual machines of pth type. Action of adjusting the resources denoted by (xi,m+n+1 xi,m+n+2 . . . xi,m+n+w), where xi,m+n+p represents the count of virtual machine of pth types which has to be adjusted. QoS value in present state is denoted by (xi,m+n+w+1) and QoS value after adjustment indicated by (yi).

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

Table 1

L	VMs	$Action$	QoS_t	QoS_{t+1}
$x_{0,0}$ $x_{0,1}$ …$x_{0,m}$	$x_{0,m+1}$ $x_{0,m+2}$ …$x_{0,m+n}$	$x_{0,m+n+1}$ $x_{0,m+n+2}$ …$x_{0,m+n+w}$	$x_{0,m+n+w+1}$	y_0
$x_{1,0}$ $x_{1,1}$ …$x_{1,m}$	$x_{1,m+1}$ $x_{1,m+2}$ …$x_{1,m+n}$	$x_{1,m+n+1}$ $x_{1,m+n+2}$ …$x_{1,m+n+w}$	$x_{1,m+n+w+1}$	y_1
…	…	…	…	…
$x_{u,0}$ $x_{u,1}$ …$x_{u,m}$	$x_{u,m+1}$ $x_{u,m+2}$ …$x_{u,m+n}$	$x_{u,m+n+1}$ $x_{u,m+n+2}$ …$x_{u,m+n+w}$	$x_{u,m+n+w+1}$	y_u

Here ISVM and DNN methods are used in training the iterative QoS model that correlates between input and output. In SVM, hyper plane equation and kernel function is set using the below formula (5):

1) Basics of Support Vector Machine (SVM)

Support vector machines (SVM) constitute two variances. Assume a vector x that has M components x_j, j =1, 2... M, i.e., M represents the dimension. The ith vector present in the dataset is denoted as x_i [24]**.** The cumulative dataset of n instances is denoted as n $\{(x_i, y_i)\}i$=1 here y_i is the label for the instance $\mathbf{x}_i$. Given data is linearly segregated, we can classify the data to two distinct classes data using the discriminant function

$$f(x) =< w, x > +b \qquad (7)$$

Where w is figured to be normal to hyper-plane, said to be the weight vector. The synonym b is termed as the bias. As per the representator theorem, discriminant function in eq. 11 is represented by,

$$f(x) = \sum_{i=1}^{n} y_i a_i < x_i, x > +b \qquad (8)$$

The kernel function is derived as:

$$k(x, x') =< x, x' > \qquad (9)$$

Data from bioinformatics symbolizes high dimensional in nature, so support vector machines is used as data classifiers. The concept behind SVM hides behind large data separation.

- **Divergence Ratio Tuning for SVM**

A support vector machine makes decisions from two various classes of the input. Most of the applications don't associate the input variable to any of a class. Divergence Ration is applied to every input and SVMs are reframed, because of which each input makes various contribution to the decision layer. So in this work proposed the Divergence Ratio Tuning for SVM .This process will be explained in the below section.

- **Relationship between Classification Data and Curvature of the Separating Hyperplane**

In the condition of lower intra-class variability and higher inter-class variability, a linear separating hyper-plane helps to split into different classes. Given by the Fig. 3. Separating hyper-planes for classes with a) high inter-class variability and low intra-class variability b) Low inter-class variability c) high intra-class variability and curved hyper-plane is necessary for class separation. The ratio of the interclass variability to intra-class variability is called as *variability ratio* and it helps to find the appropriate curvature of the separating hyper-plane, that could be manipulated by optimal changes in kernel parameters

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

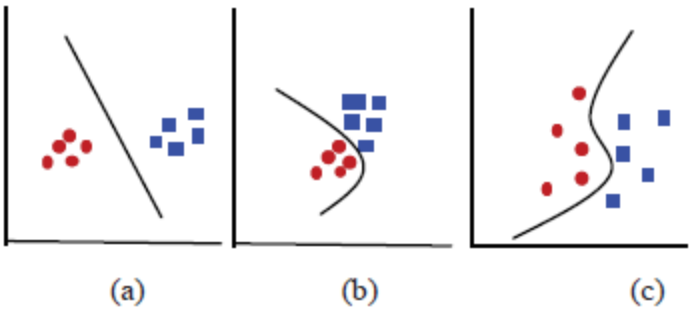

Fig. 3. Separating hyper-planes for classes with a) high inter-class variability and low intra-class variability b) Low inter-class variability c) high intra-class variability.

- **Sigmoid kernel**

The kernel is assumed as positive definite value. But the sigmoid kernel which is used widely is not positive and semi definite for some of the factors. Hence the variablesσ, r should be selected carefully, else the output could become an incorrect value and SVM performance would be bad than behaving it in random.

$$k(x_i, x_j) = \tanh(\sigma x_i^T x_j + r) \qquad (10)$$

Here σ is the scaling variable of all the input samples, r acts as a shifting variable which controls the mapping threshold (so r =0).

- **Estimation of Divergence Ratio**

Divergence Ratio is described as the intra class variability divided by the inter cluster variability. Assume x_i^A as ith n dimensional point associated with class A. Consider the average af all the points in the class to be $\overline{\boldsymbol{x}}^{\boldsymbol{A}}$. Hence the intra-class variability is termed to be ,

$$\gamma(A) = \frac{\sum_{i=1}^{k} \left| x_i^A - \overline{x}^A \right|}{k} \qquad (11)$$

Here k refers to the number of n-dimensional values associated with class A .γ (A) refers to the intra-class variability belonging to class A. The inter-class variability of two classes ***A*** and ***B*** is said to be the difference between the average values of the points associated with the respective classes. Hence inter-class variability is termed as,

$$\delta(A, B) = \left| \overline{x}^A - \overline{x}^B \right| \qquad (12)$$

Divergence Ratio of classes A and B is found by

$$\text{Divergence Ratio} = \frac{\gamma(A)+\gamma(B)}{\delta(A,B)} \qquad (13)$$

Fig. 4. Illustrates that a higher variability ratio needs a higher curvature of the separating hyper-plane.

2. Deep Neural Network (DNN)

In due of machine learning based power allocation process, supervised learning technique is used in a completely interconnected neural network. DNN comprises a single input layer, several hidden layers and an output layer [25]. Hidden neurons tend to decrease as the hidden layers increase. This leads to increase in the efficiency of the computation. Channel power gains from V2I and the V2V links is given as inputs to the DNN and the best allocation of Power is obtained as result from the DNN.

Bias inputs are not applied as inputs to neurons irrespective of layers. Rectified liner unit (ReLU) which acts as the activation function is used by hidden layers and activation function in the output layer rectifies the power constraints problem. In general, in the hidden layer, activation function is given by

$$y_{hidden} = \max(0, x_{hidden}) \qquad (14)$$

The output layer is given by

$$y_{out} = \min(\max(0, x_{out}), P_{max}^{x} \qquad (15)$$

A set of channels is given as input to the trained neural network for producing best power allocation strategy as a result. The sum rate is obtained as an average of the test data sets.

B. Energy Efficient Model using Modified Clonal Selection Algorithm (MCSA)

In the current section, resource allocation along with energy-efficient technique using Modified Clonal Selection Algorithm is explained. The parameters used in the proposed system architecture, Energy- Efficient based modified clonal selection algorithm (EE-MCSA) are detailed along with the respective assumptions and models. The challenges in managing various applications deployed in the cloud enforce efficient resource allocation and workload variance. Fig 4 depicts the energy-efficient model of resource allocation in a Cloud datacenter.

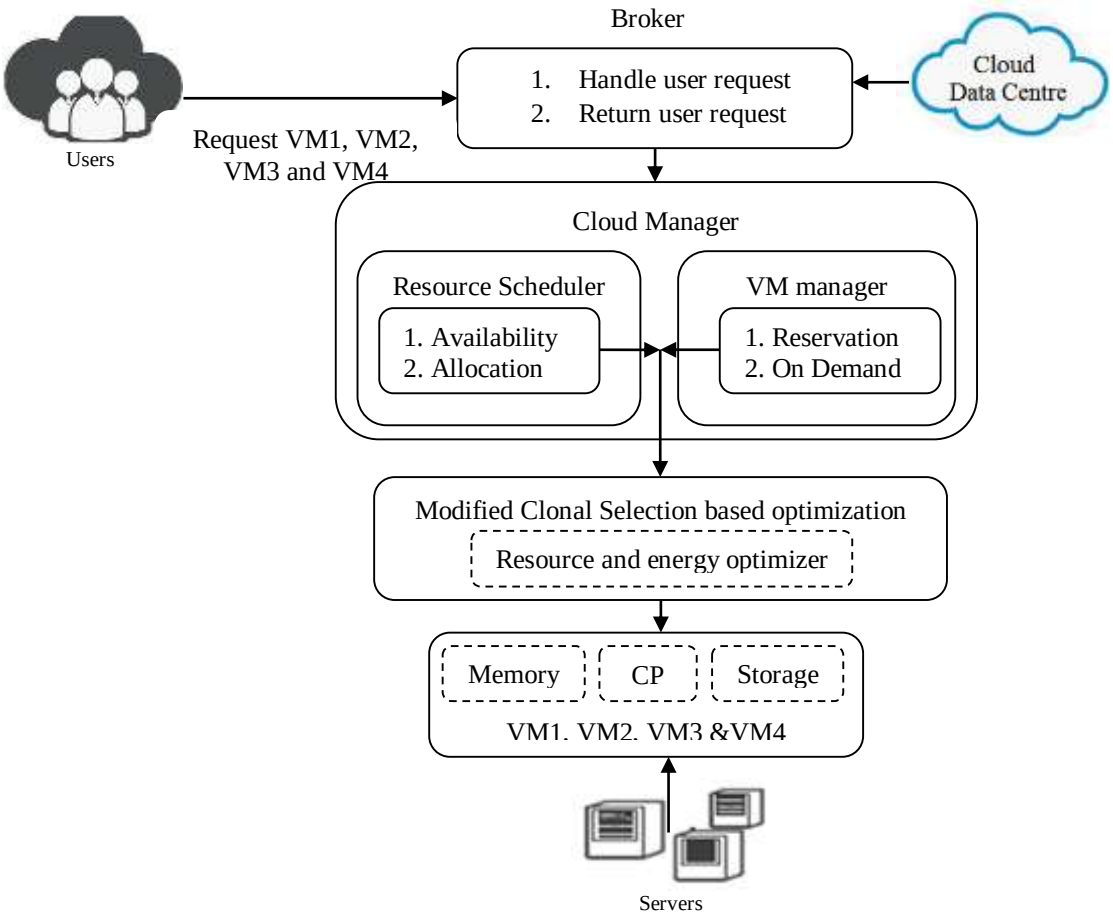

Figure.4. Energy-Efficient Model of Resource allocation

It depicts the request flow path of the user from broker to the Cloud services and then it flows to the datacenter. User submits their request initially to Cloud broker and it will send the response as per the need, and othe how the data centers perform, to which broker gets subscribed.

After the request reaches the datacenter the cloud manager is responsible for making a conclusion by checking the request with VM manager and resource scheduler module. The modules are meant to evaluate the resource parameters such as reservation, allocation, availability, demand. Cloud manager determines the request to be accepted or not based on system availability. But it did not solve the efficiency of resource allocation process which resulted in lower utilization of resource and energy management in data centers. Henceforth a new optimization method using Modified Clonal Selection algorithm is developed that overcomes the above drawback.

It is assumed that there are numerous tasks .A task have multiple subtasks that has precedence constraints. A subtask processes any resource which is currently available. Each of the cloud resource has a specified requirement (e.g., CPU, memory, network, storage). Any subtask is computed taking a single resource, and the resource allocated is made available at any point of time. The implementation of MCSA in a general process is given below:

Inputs: Consider $R = (R1, R2, \ldots, Rj, \ldots, Rm)$ as a set of m available resources that processes n independent tasks given by the set $T = (T1, T2, \ldots, Ti, \ldots, Tn)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. Resources that are available are not related and are parallel. Every task Ti could be processed using a subset $Rj \in R$ of available resources.

Outputs: A resource allocation technique which has more efficiency as a task scheduler and resource allocator.

Objectives: Upgrading the data center with more energy efficient techniques.

Real time applications use optimization principles which framed a resource allocation technique that considers two factors for energy conservation.

1) Clonal Selection algorithm (CSA)

Clonal Selection Algorithms (CSA) is the featured Immune algorithms (IA) which uses Clonal Selection Principle. It uses implementing concepts namely adaptive mutation factor [26] to enhance its performance. It manages to have a constant Factor in line with the process which is decided by the affinity of antibodies. A CLONALG algorithm works on the basis of population where its search depends on its mutation operator. The proposed work, importance is laid on the mutation operators to be more effective. The Clonal Selection Algorithm (CSA) reproduces individuals with high affinities and chooses improved maturate progenies. Hence the technique promotes a greedy search, where there is an optimization of single members and newcomers are given importance in search area. Because of this feature, CSA to be more appropriate in optimization problems.

Adaptive mutation factor based Clonal selection algorithm

As per CLONALG algorithm, best antibody is obtained based on cloning rate (β) and the optimal clones are created. Using the technique, best antibodies are mutated with worst ones. A lesser percentage of best antibodies created are mutated with the worst ones. As the difference in affinity is high, then worst antibody vanishes and will go by the optimal as like in any of the Evolutionary technique. While the affinity variation becomes low, all the worst and best antibodies occupy the same area of the search space and worst antibody comes nearer to best antibody's area. Further improvements don't achieve a rapid change. Further improvements done through adding little more antibodies in mutation. This increases the convergent rate by adaptive increment of cloned antibodies. Mutation factor are dependent on the affinity measure. As the ratio of the affinity among the best and worst antibody is lesser than threshold value (μ), mutation parameter must be increased. In this work adaptive mutation factor based CSA is proposed to reduce the premature convergence issue.

The following pseudo code is included in affinity function for avoiding the premature convergence problem of the basic CLONALG:

$$If \left(\frac{aff[ab_b]}{aff[ab_w]}\right) < \mu \qquad (16)$$

$$\delta = \delta \times \rho \qquad (17)$$

Where ρ refers to the parameter, which varies as shown below

$$\rho = \left(\frac{iter}{maxiter}\right) * \alpha \qquad (18)$$

Here:

μ represents threshold value,

ab_b refers to the best antibody in the pool of Antibodies.

ab_w refers to Worst antibody in the Antibody Pool.

α represents Constant multiplier based on the problem type.

δ denotes Adaptive Mutation Factor.

Iter represents present Iteration.

Maxiter denotes Maximum number of Iterations.

Aff(.) Represents the function for finding the affinity of the antibody.

Algorithm 1: Modified clonal Selection algorithm (MCSA)

Input: No. of. User requests and mutation probabilities P_m.
Output: Individual that has minimal objective function value

1. Randomly generate an antibody population A(0)
2. Calculate affinity of the initial population A(k)
3. Choose half of the antibodies that has greater affinity as the population A_l(k)
4. Clone each individual in A_l(k) to create the population B(k), and the clonal number is proportional to their affinity
5. Do mutation from the population A_l(k) to form the population C(k)
6. Evaluate individual affinity through adaptive affinity function based mutation using eq. (16, 17). Recalculate of affinity values for new mutated antibodies.
7. Perform selection operation from the population C(k) and obtain the next generation population.

For energy efficient based resource allocation, as soon as the request falls system invokes the MCSA to adjust the resource allocation. Mappings between tasks and resources into binary value defined to be a collection of initial population $X(0)$ are modified to select the optimum solution in MCSA Any individual is represented as $X_i^G = (X_{i1}^G, X_{i2}^G, \dots\dots\dots X_{ip}^G,)$, Here *G* indicates the present generation, $i = 1, 2, \dots, s$, where s represents the population limit.

Every individual (antibody) is named as a candidate that is given by a binary string of bits. User selects a bit string for finding an appropriate result for an issue. Every gene in chromosome is set as 0 or 1. As soon as the initial population is created, affinity value of an individual is estimated, saved to be used in near future. MCSA is

implemented to allocate resources and for optimization problems. The affinity function depends on energy efficiency. Affinity function is described as below:

$$aff(x) = e^{minE_i + minMs} \quad (19)$$

1) **Minimizing Energy Consumption**

Mathematically, resource allocation model is represented in a formula as given by Eq. (20).

$$\sum_{x=1}^{Z,n}(PC_i + PD_i + PM_i) \times T_i \rightarrow U_i^j \quad (20)$$

Datacenters present in the cloud provide virtual resources of count m, $V = (V_1, V_2, V_3, \ldots . V_m)$ into n physical resources used by the datacenter through a restricted resource function through $P = (P_{c1}, P_{c2}, P_{c3}, \ldots . P_{cn})$, $P = (P_{d1}, P_{d2}, P_{d3}, \ldots . P_{dn})$ and $P = (P_{m1}, P_{m2}, P_{m3}, \ldots . P_{mn})$ for the consumers of the Cloud services $U = (U_1, U_2, U_3, \ldots . U_n)$ and the fitness of j objective function $F = (F_1, F_2, F_3, \ldots . F_z)$ are maximized.

The prime motive of this research is reducing the energy utilization in a data center used in cloudlets with respect to the to request volume .Precisely, it is the total power required to deploy the host or PMs in datacenters It is given by the Eq. (21)

$$Energy\ Cons = \sum_{source} i \int_{str_{time}}^{fns_{time}} E_i(F, T) \quad (21)$$

The energy consumption for a resource i at time T and placement F. The Ei denotes the consumption of energy by resource i that is calculated from the initial time to end time of utilizing it.

2) Maximizing Resource Utilization

Resource Utilization Model helps in finding the exact number of resources used in the data centers. Hence, Cloud manager needs incorporating a solution for efficient allocation of requests from the resource pool to meet the need of a user in the data center. The resource utilization of a physical host is represented and illustrated through Eq. (22).

$$Resource\ Utiliz = \frac{\sum_{resource} i\ execution_{time}}{makespan\ or\ maxtask^i(i\ execution_{time})} \quad (22)$$

Here makespan is the total completion time taken once resource allocation done to the users and execution time is calculated as a difference between the requests from the initial and execution time of the requests.

C. On-line decision-making based on ICSA algorithm

The current section defines the Self-adaptive Management module. It describes about the MCSA algorithm which is used to develop an innovative resource allocation technique according to QoS prediction model [27]. This will improve the traditional CSA, ICSA version will replace the original cuckoo's movement by a new strategy which includes the new concept of diversity function will explained in the below section.

1) Diversity function based Cuckoo Search algorithm (ICSA)

CSA algorithm which is based on cuckoo's concept of breeding and adaptation comes with following assumptions [26]:

(i) Every cuckoo lays a single egg in a nest which is selected at random .Here the egg denotes a solution for a problem which is in study.

(ii) It uses survival of the fittest strategy. High quality eggs in the fittest nest are then sent to next generation.

(iii) Host nests are decided in advance. The bird which is a host finds the egg through probability. In such way the bird expels the egg or disregards the nest and search a new place in order to reconstruct a nest.

CS, another nature-enlivened algorithm dependent on the commit brood parasitic conduct of some cuckoo species in mix with the Levy flights conduct of certain flying creatures and organic product flies is a straightforward yet encouraging populace based stochastic hunt. In common, a nest signifies a candidate solution $X = (x_1,...,x_D)$, while processing a objective function $f(x)$ along with solution space $[x_{j,\min}, x_{j,\max}]$, $j = 1, 2, \ldots, D$. Just like Classical evolutionary methods, iteration process in CS consists of initial and evolutional phase.

Initial phase consists of whole population called solution that is randomly sampled from solution space through a formula

$$x_{i,j,0} = x_{i,j,min} + r\left(x_{i,j,max} - x_{i,j,min}\right), \qquad i = 1,2,\ldots.N, \quad (23)$$

CS performs two random walks in iteration: Levy flights random walk (LFRW) ,biased random walk (BRW) for finding unique solutions. LFRW is a random walk; step size of it is derived from Levy distribution. In generation G ($G > 0$), LFRW can be represented by a formula

$$x_{i,G+1} = x_{i,G} + \alpha \oplus Levy(\beta), \quad (24)$$

Here α refers to the step size associated with the scales of the problem. $\oplus$ represents entry-wise multiplications. Levy (β) is derived from a Levy distribution and given for large steps as:

$$Levy(\beta) \sim u = t^{-1-\beta}, 0 < \beta \leq 2. \quad (25)$$

In CS, LFRW is utilized to find for new result around the best solution acquired up until now and executed by the accompanying formula

$$x_{i,G+1} = x_{i,G} + \alpha_0 \times \frac{\phi \times u}{|v|^{1/\beta}} \times \left(x_{i,G} - X_{best}\right), \quad (26)$$

where α_0 denotes a factor ($\alpha_0 = 0.01$) and X_{best} denotes the best solution got so far,

$$\phi = \left(\frac{\Gamma(1+\beta) \times \sin((\pi \times \beta)/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}}\right)^{1/\beta}, \quad (27)$$

Where β denotes the constant and given as 1.5, u and V denote the random numbers derived from a normal distribution with mean of 0 and standard deviation of 1, and Γ denotes the *gamma function*.

BRW helps to find unique ways by using randomization technique that is better than the present techniques the first one creates a trial one using mutation of the current solution as base vector and two randomly selected

solutions as perturbed vectors. The other one uses distinct solution created using crossover operator from the current and the trial solutions. BSRW can be formulated as follows:

$$x_{i,j,G+1} = \begin{cases} x_{i,j,G} + x_{m,j,G} - x_{n,j,G}), & if\ r_a > p_a, \\ x_{i,j,G} & otherwise, \end{cases} \quad (28)$$

Here random indexes m and n denote the m^{th} and n^{th} solutions in the population and , j denotes the j^{th} dimension of the solution, r and r_a denotes the random numbers in the range [0, 1], and p_a is a fraction probability.

Once the random walk gets completed, CS finds a improved technique based on the generated and the current solutions fitness that uses greedy technique. After each iteration, the best solution is derived as in algorithm 2.

The main difference among the ICSA and CS lies in the way how we the parameters p_a and α are adjusted. In order to reform the efficiency of the CS algorithm and to overcome the shortfalls of using those fixed parameters the ICS algorithm makes use of parameters p_a and α. Traditional methods use large values for p_a and α to implement the algorithm and to improve the solution efficiency. But these parameters must be decreased in order to get a optimal value for solution vectors. The parameter values p_a and α are modified in regard to the number of generations and are given by the Equations 41-43, here NI and gn represent the number of total iterations and the current iteration.

$$p_a(gn) = p_{amax} - \frac{gn}{NI}(p_{amax} - p_{amin}) \quad (29)$$

$$\alpha(gn) = \alpha_{max}\exp(c.gn) \quad (30)$$

$$c = \frac{1}{NI} Ln\left[\frac{\alpha_{min}}{\alpha_{max}}\right] \quad (31)$$

C is the diversity function for tuning the $p_a and\ \alpha$.

Input: No. of. User Request
Output: Allocated Resources
(1) Initialize the parameter values generate the random initial vector values and set the iteration number t = 1.
(2) Evaluate fitness values of each individual and determine the current best individual with the best objective value. Verify if the stopping criterion is met. If the stopping criterion is met, then output the best solution; else update the iteration number $t = t + 1$ and repeat the iteration process until the stopping criterion is met.
(3) Keep the best solution of the last iteration, and get a set of new solutions $Xnew = \left[x_1^{(t+1)}, \ldots., x_i^{(t+1)}, \ldots\ x_K^{(t+1)}\right] by$ L´evy flight,
(4) Evaluate the fitness value $F_i^{(t+1)}$ of the new solution $x_i^{(t+1)}$, and compare $F_i^{(t+1)}$ with $F_i^{(t)}$ that signifies the solution of the t^{th} iteration.
(5) A fraction (p_a) of worse nests is abandoned and new ones are created.
(6) Implement the orthogonal design strategy procedures.
(7) Keep the best solution.
(8) Search for a new solution using diversity function based using Eqs.31
(9) Keep the best nest with quality solution
(10) Rank the nests and find the current best one
(11)Pass the current best nest to the next generation
(12) Go to step (2)
End

IV. RESULTS AND DISCUSSION

Experiments are conducted on Cloud Stack with three forms of virtual machines categorizing small, medium and large. The objective for evaluating it is to 1) Verify the iterative QoS model with conventional machine learning models for its accuracy and with the historical data.2) Evaluate how resource allocation performs by verifying it with the output allocated using energy efficient processes. The below figures help us to convey how the proposed resource allocation technique performs.

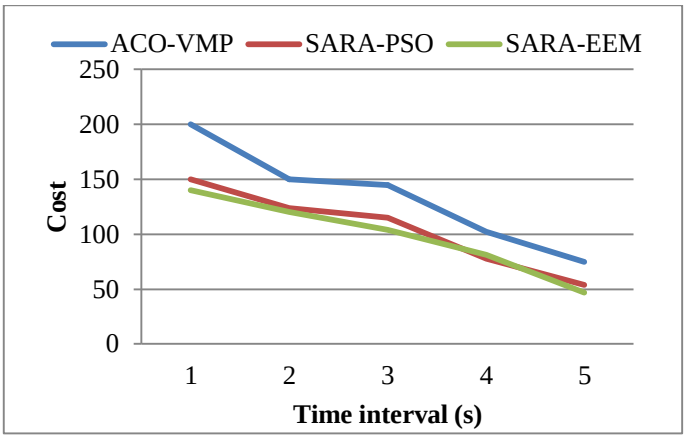

Figure. 5. Cost effectiveness - Resource allocation techniques (Proposed Vs Existing)

Fig 5 shows the balance chart of cost Vs QoS of the above mentioned techniques. For example the proposed technique has a QoS and Cost value that lies between SARA-PSO technique and ACO-VMP technique taken from 1 to 3 time intervals. Also at 4^{th} and 5^{th} intervals the proposed technique does not perform well with low value of QoS but with low cost. Hence it could be taken into matter that proposed technique ACO-VMP method is more cost effective.

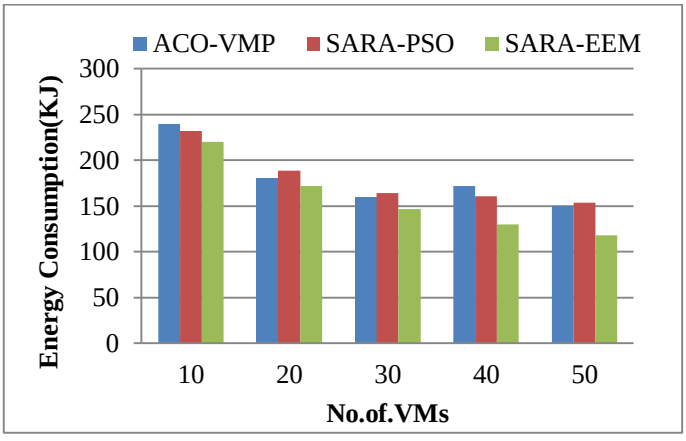

Figure. 6. Energy Consumption range- (Proposed Vs existing resource allocation techniques)

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

Fig.6. illustrates the comparison of energy chart between proposed and existing resource allocation techniques. The proposed SARA-EEM technique outperforms SARA-PSO and ACO-VMP. SARA-EEM presents guaranteed of convergence using Improved Cuckoo Search algorithm (ICSA). But because of the changes in the diversity function parameters, the convergence time is an issue factor. Experiments reveal that the proposed SARA-EEM method needs less than a min to sort out the allocation issue. This occurs since the QoS prediction model is more flexible in finding a optimal value. Since it converges at a faster rate, the proposed technique could be applied to data centers.

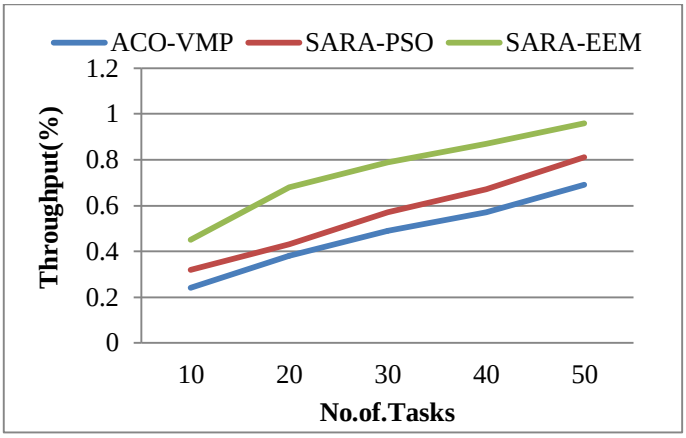

Figure. 7. Performance Throughput - (proposed Vs existing) resource allocation techniques

The throughput is compared to the traditional approach SARA-PSO and ACO-VMP. Throughput has a direct effect in allocating resources and the proposed work predictions are more accurate as given in Figure 7.Also it states that the resources are utilized high in the proposed work when verified with the existing methods. So there is a low energy conservation which allocates most of the VMS and hence the efficiency is high. Hence SARA-EEM performs best than SARA-PSO and ACO-VMP in the context of energy and resource utilization.

V. CONCLUSION

This work uses iterative QoS prediction model, an Energy Efficient Model based on Modified Clonal Selection Algorithm (MCSA) and Improved Cuckoo Search Algorithm (ICSA) .It is a self-adaptive technique for allocating resources in cloud environment. In this work, two prediction models of QoS works by iteration are designed using ISVM and DNN. Then the Energy efficient model is done by using MCSA. Finally the resource allocation decision criteria are satisfied by using ICSA. Cloud environment is used to simulate and test the proposed work. Experiments indicate that the proposed Energy efficient and self- adaptive framework has more accuracy rate compared to the older QoS algorithms. The proposed SARA-EEM framework is effective by consuming less energy and optimized resource usage in comparison to SARA-PSO and ACO-VMP.

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

REFERENCES

1. Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P. P., Kolodziej, J., Balaji, P., ... & Khan, S. U. (2016). A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing*, *98*(7), 751-774.
2. Shyamala, K., & Rani, T. S. (2015). An analysis on efficient resource allocation mechanisms in cloud computing. *Indian Journal of Science and Technology*, *8*(9), 814.
3. Buyya, R., Beloglazov, A., & Abawajy, J. (2010). Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. *arXiv preprint arXiv:1006.0308*.
4. Mohan, N. R., & Raj, E. B. (2012, November). Resource Allocation Techniques in Cloud Computing--Research Challenges for Applications. In *2012 fourth international conference on computational intelligence and communication networks* (pp. 556-560). IEEE.
5. Banerjee, A., Agrawal, P., & Iyengar, N. C. S. (2013). Energy efficiency model for cloud computing. *International Journal of Energy, Information and Communications*, *4*(6), 29-42.
6. Li, H., Zhu, G., Cui, C., Tang, H., Dou, Y., & He, C. (2016). Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing. *Computing*, *98*(3), 303-317.
7. Parikh, S. M. (2013, November). A survey on cloud computing resource allocation techniques. In *2013 Nirma University International Conference on Engineering (NUiCONE)* (pp. 1-5). IEEE.
8. Yang, Z., Liu, M., Xiu, J., & Liu, C. (2012, November). Study on cloud resource allocation strategy based on particle swarm ant colony optimization algorithm. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems* (Vol. 1, pp. 488-491). IEEE.
9. Shiny, J. J., & Vignesh, S. (2017, January). A comprehensive review on QoS measures for resource allocation in cloud environment. In *2016 Eighth International Conference on Advanced Computing (ICoAC)* (pp. 157-164). IEEE.
10. Li, Y. K. (2014). QoS-aware dynamic virtual resource management in the cloud. In *Applied Mechanics and Materials* (Vol. 556, pp. 5809-5812). Trans Tech Publications Ltd.
11. Kumar, N., & Saxena, S. (2015). A preference-based resource allocation in cloud computing systems. *Procedia computer science*, *57*, 104-111.
12. Goudarzi, H., & Pedram, M. (2011, July). Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems. In *2011 IEEE 4th International Conference on Cloud Computing* (pp. 324-331). IEEE.
13. Wang, H., Tianfield, H., & Mair, Q. (2014). Auction based resource allocation in cloud computing. *Multiagent and Grid Systems*, *10*(1), 51-66.
14. RamMohan, N. R., & Baburaj, E. (2014). Resource Allocation Using Interference Aware Technique in Cloud Computing Environment. *International Journal Of Digital Content Technology And Its Applications*, *8*(1), 35.

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020

15. Xiong, A. P., & Xu, C. X. (2014). Energy efficient multi resource allocation of virtual machine based on PSO in cloud data center. *Mathematical Problems in Engineering*, *2014*.
16. Sharma, N. K., & Reddy, G. R. M. (2015, March). Novel energy efficient virtual machine allocation at data center using genetic algorithm. In *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)* (pp. 1-6). IEEE.
17. Wang, S., Liu, Z., Zheng, Z., Sun, Q., & Yang, F. (2013, December). Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers. In *2013 International Conference on Parallel and Distributed Systems* (pp. 102-109). IEEE.
18. Liu, X. F., Zhan, Z. H., Du, K. J., & Chen, W. N. (2014, July). Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach. In *Proceedings of the 2014 annual conference on genetic and evolutionary computation* (pp. 41-48).
19. Joseph, C. T., Chandrasekaran, K., & Cyriac, R. (2015). A novel family genetic approach for virtual machine allocation. *Procedia Computer Science*, *46*, 558-565.
20. Tang, M., & Pan, S. (2015). A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. *Neural processing letters*, *41*(2), 211-221.
21. Marphatia, A., Muhnot, A., Sachdeva, T., Shukla, E., & Kurup, L. (2013). Optimization of FCFS based resource provisioning algorithm for cloud computing. *IOSR Journal of Computer Engineering (IOSR-JCE).(Mar.-Apr. 2013)*, *10*(5), 1-5.
22. Srinivasa, K. G., Srinidhi, S., Kumar, K. S., Shenvi, V., Kaushik, U. S., & Mishra, K. (2014, February). Game theoretic resource allocation in cloud computing. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)* (pp. 36-42). IEEE.
23. Chen, X., Wang, H., Ma, Y., Zheng, X., & Guo, L. (2020). Self-adaptive resource allocation for cloud-based software services based on iterative QoS prediction model. *Future Generation Computer Systems*, *105*, 287-296.
24. Andrews, S., Tsochantaridis, I., & Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Advances in neural information processing systems* (pp. 577-584).
25. Canziani, A., Paszke, A., & Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.
26. De Castro, L. N., & Von Zuben, F. J. (2000, July). The clonal selection algorithm with engineering applications. In *Proceedings of GECCO* (Vol. 2000, pp. 36-39).
27. Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)* (pp. 210-214). IEEE.

Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020