International Journal of Engineering & Technology, 7 (4) (2018) 3530-3537

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.17859
Research paper

Application Aware Energy and Cost Efficient Resource Provisioning in Cloud

Shreenath Acharya [1] *, Demian Antony D'Mello [2], Raghavendra Achar [3]

[1] Computer Science & Engineering Department, St Joseph Engineering College, Vamanjoor, Mangaluru
[2] Head of Computer Science & Engineering Department, Canara Engineering College, Benjanapadavu, Mangaluru
[3] Information & Communication Technology Department, Manipal Institute of Technology, Manipal, Udupi
**Corresponding author E-mail:shree.katapady@gmail.com*

Abstract

The high popularity and growing demand of cloud computing has a strong effect on the cloud infrastructure providers to efficiently manage their cloud datacenters in order to fulfill provisioning of everything in the form of a service to end users and also to achieve efficient balancing between its less energy consumption for reduced environmental affects and maximize revue. This paper presents an energy efficient framework for green cloud datacenter which considers resource utilization and energy efficiency to support resource allocation decisions towards green computing. This work mainly relies on energy efficient provisioning of resources utilizing an application prediction and VM provisioning mechanism using genetic algorithm. Our approach has been validated by performing a set of experiments under dynamic cloud environment workload scenarios using Cloudsim toolkit. Compared to the benchmark (existing) algorithms, our method is able to significantly reduce the energy consumption cost without a priori knowledge of the future workloads

Keywords:*Cloud Computing; Energy Efficient; Migration; Resource Utilization; Virtual Machine.*

1. Introduction

Cloud computing is a modern style of ubiquitous computing providing on-demand access to a shared pool of configurable resources over the internet through providers' datacenters. Resource utilization is the key technique in cloud computing which enables the utilization of the computing resources to facilitate the execution of the complex tasks. It considers many important factors like response time, load balancing and energy consumption. Selection of a node capable of executing a given task requires satisfying the quality of service (QoS) specifications by users through service level agreements (SLA) and also minimizing energy consumption [1].

The cloud providers need to ensure the on-demand QoS with increased utilization of resources [2]. It requires resource allocation in a fine grained manner according to the applications demand. The necessary precondition for resource allocation may be by predicting future load in advance based on the some predefined logic.

The vast amount of datacenters hosted by leading service providers like Microsoft, IBM, Google and Yahoo are required to provide sufficient measures to limit the energy consumptions of applications which contribute to high operational cost and carbon footprints in the environment. Thus green computing solutions needed to reduce the operational costs along with environmental effects [1][16]. Green computing necessitates the reduction of energy consumption which in turn depends upon server consolidation and migrations. The pressure from the governments worldwide also aiming to reduce carbon footprints from the datacenters impacting climate change. The growing demand of data and computing applications makes it a very challenging task to provision large servers and associated disk requirements to process them within the stipulated time.

In business application process for green computing, allocation of the resources should take into account main factors like energy consumption and the makespan. The efficiency of the resource allocation to execute the tasks is mainly in improved energy efficiency of the datacenter. Green computing mainly achieves efficient utilization of the resources and also results in reduced energy consumption.

For ex: Consider a situation wherein the jobs/tasks/applications are (CPU Intensive, Memory Intensive and I/O Intensive) assigned to a cloud datacenter randomly at different intervals of time. In this case, the tasks are assigned to the hosts without considering the capacity of hosts/servers, which leads to more number mismatches in the resource availability to fulfill the request asking for more VM migrations. More number of migrations, lead to increase in the response time thereby resulting in performance degradation. It also results in more energy and cost incurred because of the lower utilization of the resources.

CPU Intensive means video streaming applications requiring more speedy processing (like having more number of cores). I/O means web applications requiring more disk space and communications. Memory intensive means applications requiring more memory and CPU speed.

The main objective of this paper is optimized resource allocation using genetic algorithm for the selection of hosts based on execution time and power/energy consumption models in a cloud computing environment. It also uses prediction of the tasks before submission for processing. The proposed mechanism shows improved effectiveness in resource allocation compared with existing algorithms. The results obtained after the experiments using CloudSim toolkit shows that the proposed scheme has resulted in

Copyright © 2018Shreenath Acharya et. al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

enhanced energy efficiency while satisfying the Service Level Agreements by the users/consumers.

The specific contributions of this paper includes the following:

- A literature survey about various existing energy efficient resource allocation algorithms and an analysis of their advantages and disadvantages are presented.
- An effective energy-efficient optimization model for resource allocation in cloud computing environments is proposed.
- An algorithm for resource provisioning in cloud computing environments inspired by genetic algorithm is proposed.
- Performance analysis of the proposed algorithm and an evaluation of the algorithm with respect to other existing algorithms are presented.

The rest of this paper is organized as follows: Section 2 discusses related works, followed by models for energy-efficient optimization and makespan optimization design in Section 3. The improved selection algorithm for resource allocation is discussed in Section 4. Section 5 shows the simulated experimental results, and Section 6 concludes the paper with summary and future research directions.

2. Literature survey

There has been a lot of research carried out by many researchers specifically in the area of scheduling, load balancing, VM provisioning and energy efficiency. MyintMyatMyo et al. [3] have proposed an energy efficient resource allocation framework which enables automatic allocation with an aim to drop the total cost of ownership(TCO) for the providers with reduced violation of SLA and consumption of the energy. To achieve this, they have utilized, reinforcement learning (RL) approach for handling dynamic variation of unpredictable workloads through automated environmental sensing. But their approach leads to more complexity as per the increase in the number of resources.

Wanneng Shu et al. [4] have presented a clonal optimization based novel energy efficient algorithm to achieve green computing. Their approach called Improved Clonal Selection Algorithm (ICSA) mainly considers the energy consumption and makespan as a prime factor of consideration to achieve their objective. They have developed an optimization model for energy based on dynamic voltage frequency scaling (DVFS) approach, makespan optimization model by considering the overall time taken for processing the tasks and a multi-objective optimization model utilizing both energy and makespan for resource allocation. They have not considered all the operators and computation complexity in order to match more with practical configurations/set up.

A statistical learning based future load forecasting strategy (KSwSVR) [5] to improve the resource utilization and ensure QoS while delivering the services. The authors approach utilizes integrated version of support vector regression and kalman filter. Their improved support vector regression (SVR) algorithm considers more weights given to critical data than the traditional SVR thereby utilizing the historical information effectively. The kalman filter reduces the noise from the data to result in better accuracy. This approach resulted in prominent energy savings as well as better resource utilization thereby meeting the desired SLAs.

A fuzzy logic based energy efficient load balancing algorithm [6] is implemented based on the renewable energy sources in order to solve the problem with unpredictable workload and energy costs in cloud environments. They have modelled it based on the cloud providers datacenters spaced at multiple locations across the globe by considering their energy consumption by way of local on-site production using renewable sources as well as from utility grid. This Geographical load Balancing (GLB) algorithm considers the user request to be processed by identifying a datacenter which has higher renewable energy to be sufficient to fulfill the task thus ensuring reduced overall cost. Through experimentation on real-world traces they have proved the effectiveness of this approach in reducing electricity costs by way of reduced energy consumption and better utilization of resources as well as the datacenters.

R. R. Darwish [7] has presented autonomic cloud resource orchestration architecture to minimize power consumption for web applications workload. The architecture contains global and local controllers using heuristic as well as control theory approaches to fulfill the objective. Global controller utilizes heuristic method ie, bees algorithm for mapping the VM to the appropriate resources. Local controller follows proactive fuzzy controller based strategy in order to sustain from the workload fluctuations. Through simulations they have shown that the response time is better than the Queuing-Theoretic Feed Forward based Predictor (QFF) at higher arrival rates. And also the energy consumption and SLA violations are considerably less.

Sanket Dangi et al. [8] have proposed a self-tuning energy aware model for server clusters to reduce the energy consumption by forcing the servers in a cluster to hibernate mode while they are in idle state. They have used historical data of network workload to take (make) decisions about which servers to be hibernated. The load balancer has been configured to move the tasks from one server to another only when it is full. The server cluster follows a master slave approach wherein a master server performs all the operations such as accepting user requests, load distribution, storing network traffic details and recognizing patterns from it, reconfiguring clusters and the request forwarding to active servers.

Anton Beloglazov et al. [9] have proposed an adaptive threshold based energy efficient strategy to overcome the trade-off between energy and performance. This model contains a global manager, number of local managers based on the number of nodes and a virtual machine monitor (VMM). The global manager collects information from all the local managers in order to make decisions and issue commands about switching off the idle nodes, switching them to sleep/hibernate. The VM placement is considered like a bin packing problem and a modified best fit decreasing (MBFD) algorithm has been utilized which allows to select the most efficient node for improving power efficiency. The simulations carried out using real workload traces from Planet lab have proved that this approach outperforms other migration-aware polices in terms of SLA violations, number of VM migrations while providing a similar level of energy consumption.

Sukhpal Singh et al. [10] have developed an energy efficiency based resource scheduling framework (EBERSF) which considers the synergy between datacenter infrastructures such as software, hardware and performance. The Green Service Allocator (GSA) coordinates with various entities like resource manager, SLA analyzer, energy manager, resource scheduler and cloud workload scheduler in order to meet the desired objective. The authors have validated their framework by considering the 3 cases as low resource usage, high resource usage and random resource usage. But, they have not incorporated any measures to improve the QoS.

Rodrigo N. Calheiros et al. [11] have developed an analytical performance based adaptive comprehensive provisioning strategy to achieve the QoS targets of the applications. Here the comprehensive provisioning contained 2 phases like VM provisioning and application provisioning. VM provisioning instantiates the virtual machines based on the hardware and software specifications of the applications. Application provisioning deploys the specialized applications like ERP, web servers within VMs and maps user requests to the application instances. They have not considered the energy and cost parameters instead concentrated more on the QoS.

Hongjian Li et al. [12] have proposed a Particle Swarm Optimization (PSO) based multi-resource based dynamic energy efficient consolidation strategy with quality of service (QoS) guarantees. The authors have considered utilization of the CPU and disk as energy efficiency metrics. The evaluation of energy efficiency has been carried out by measuring the total Euclidean distance of all the active physical machines at any point of time during the execution. The energy consumption and the VM migrations are less in their approach compared to the Modified Best fit Decreasing (MBFD) approach. But the authors have not considered the cost factor in their approach to deal with the economic benefits.

Ali Al-maamari et al. [13] have proposed a hybrid Particle Swarm Optimization (PSO) algorithm named MDPSO with combination of Dynamic PSO and cuckoo search providing significant improvements in response time and resource utilization compared to PSO and DPSO. This mechanism proved to be efficient, but there is no consideration for the factors like energy consumption, cost and load balancing.

Elina Pacini et al. [14] have implemented a dynamic scheduling algorithm based on PSO by considering number of users/jobs and the number of VMs available for execution of the tasks. Through simulations with jobs from real scientific problems they have proved that it results in increased performance than random assignment and b=genetic algorithm based scheduling. The authors have not discussed about the resource utilization, energy efficiency and the cost concerns.

Raghavendra et al. [17] have implemented an application nature aware VM provisioning architecture using genetic algorithm to predict the applications usage and appropriately provision the VMs. This resulted in lesser no. of migrations as well as SLA violations, but the authors have not considered energy and cost factors as well as resource utilization factors into account.

Shreenath Acharya et al. [18] have proposed a dynamic load balancing algorithm for resource provisioning which would perform better interms of response time and utilization. But, they have not considered energy and cost factor into account during their experimentation with varieties of servers.

3. System architecture

A novel framework of the system is shown in figure1. It contains 4 main components namely, application predictor, cluster controller, VM provisioner and migration manager.

Application Predictor: It receives the user requests in the form of tasks or applications to be executed. i.e., A = {a_1, a_2, a_3,…,a_n}.The jobs are queued for a regular interval of time and are filtered to form the task groups for allocation to the individual clusters.

Filtering process is based on the cloudlet length and its requirements and a grouping of similar requests (with some approximations) are made. Based on the grouping, applications are assigned to the clusters using first fit option.

Cluster creation is done based the individual hosts MIPS, Disk Capacity, RAM rating and No. of Cores. This results in reduced migration to result in less response time (SLA compliant). It also helps to improve system utilization which improves energy efficiency and hence cost reduction.

Job prediction is performed based on double exponential smoothing time series prediction mechanism.

Cluster Controller: It is the main controller of the individual clusters and it initiates the migration process as per the instructions from the migration manager for the overall performance of the system. It forwards (send) the task groups to the clusters according to their requirement and monitors the execution status through the Energy monitor and cost monitors while making the decisions about migration/reconfiguration of VMs.

Energy Monitor: It monitors the status of power consumption so that it does not exceed beyond the limit by way of utilization of resources.

Cost Monitor: It works in conjunction with the energy monitor and predicts the cost incurred based on it while fulfilling the SLA compliance for the applications under execution.

Migration Manager: This manager is responsible for overall migration process from one cluster to another after getting the instructions from the individual cluster managers of the clusters. It communicates with the cluster controller for informations about the performance optimizations/monitoring before identification of the best possible migration.

The cloud environment initiated migration will consider migration from one cluster to another through autonomic actions initiated from itself through the cloud migration manager.

Resource Pool consists of clusters of nodes containing system with low, medium and higher capacities of execution with appropriate processors and the devices. Each cluster is managed by a cluster manager for overall decisions about execution of assigned jobs.

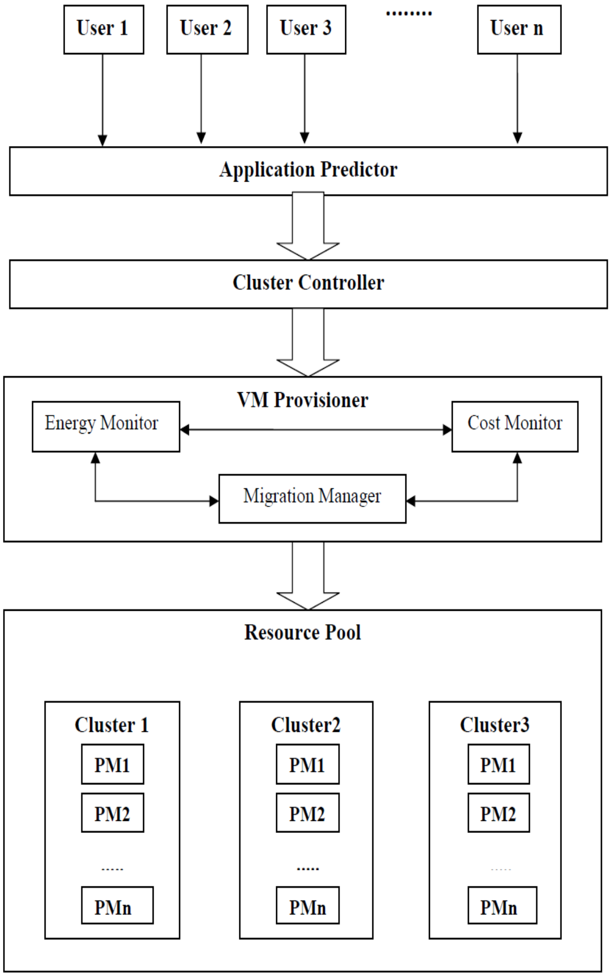

Fig. 1: Framework for VM Provisioning.

Application prediction based on history:

Double Exponential Smoothing:

In order to predict the nature of applications and the resources required, exponential smoothing prediction model has been utilized. Single exponential smoothing utilizes an exponential coefficient α, which would be inefficient to predict optimum future trend. Hence double exponential smoothing is preferred having 2 coefficients α and γ resulting in efficient and accurate prediction. This method works with identication of smoothed value and a trend estimate based on it as shown in Equation [1] and [2].

$$S_t = \alpha * A_t + (1 - \alpha)(S_{t-1} + b_{t-1}) \qquad 0<\alpha<1 \qquad (1)$$

$$b_t = \gamma (S_t - S_{t-1}) + (1 - \gamma) b_{t-1} \qquad 0<\gamma<1 \qquad (2)$$

S_t is the smoothed value for time t
b_t is the best estimate of the trend at time t
α is the data smoothing factor
γ is the trend smoothing factor
The Predicted value for a single period is calculated as,

$$F_{t+1}=S_t + b_t \qquad (3)$$

$$F_{t+1} = \alpha * A_t + (1 - \alpha)(S_{t-1} + b_{t-1}) + \gamma (S_t - S_{t-1}) + (1 - \gamma) b_{t-1} \qquad (4)$$

Substituting the values of S_{t-1} and b_{t-1} in equation 4 we get,

$$F_{t+1} = \alpha * A_t + (1 - \alpha)(\alpha * A_{t-1} + (1 - \alpha)(S_{t-2} + b_{t-2}) + \gamma (S_{t-1} - S_{t-2}) + (1 - \gamma) b_{t-2}) + \gamma (S_t - S_{t-1}) + (1 - \gamma)\gamma(S_{t-1} - S_{t-2}) + (1 - \gamma) b_{t-1} \quad (5)$$

It is evident after simplification ofEquation(5) the value ofα and γ vary as α, α(1-α), $(1-\alpha)^2$ and γ, γ(1-γ), $(1-\gamma)^2$ etc. with each term givinga clear indication that as the value of these two are high, the contribution from the 2nd, 3rd and others higher factors are less. Whereas, when these two values are less, the contributions from the succeeding terms are more.

The selection of the smoothing and the trend factors are carried out by varying the value of α and γ to identify the optimum value of them by applying the load for some duration(for ex: 5, 10 etc..). In order to evaluate our prediction model data collected from the CoMon project is utilized. We have used cpu utilization data collected for every five/ten minute from a virtual machine. The average of CPU utilization is computed for every hour. We calculate the mean percentage prediction errors, in order to evaluate the effectiveness and accuracy of this strategy.

Mean Prediction Error (MPE) =

$$\frac{|Actual\ value - Predicted\ value|}{Actual\ value} \text{ X } 100 \quad (6)$$

The different values of α and γ and its corresponding MPE for a sample set of inputs are shown in table 1.

Table 1: Mean Percentage Error for Various α and γ

α	γ	MPE	α	γ	MPE
0.1	0.1	10.08	0.6	0.1	9.12
0.2	0.2	12.05	0.7	0.2	10.82
0.3	0.3	14.18	0.8	0.3	12.79
0.4	0.4	16.55	0.9	0.4	15.07
0.5	0.5	18.92	0.1	0.6	22.99
0.6	0.6	21.33	0.2	0.7	25.64
0.7	0.7	23.99	0.3	0.8	28.30
0.8	0.8	46.43	0.4	0.9	30.95
0.9	0.9	47.70	0.5	0.1	9.18
0.1	0.2	12.30	0.6	0.2	11.07
0.2	0.3	14.48	0.7	0.3	13.03
0.3	0.4	16.85	0.8	0.4	15.37
0.4	0.5	19.22	0.9	0.5	17.74
0.5	0.6	21.66	0.1	0.7	25.98
0.6	0.7	24.32	0.2	0.8	28.63
0.7	0.8	26.97	0.3	0.9	31.28
0.8	0.9	29.63	0.4	0.1	9.35
0.9	0.1	9.19	0.5	0.2	11.31
0.1	0.3	14.78	0.6	0.3	13.29
0.2	0.4	17.15	0.7	0.4	15.66
0.3	0.5	19.52	0.8	0.5	18.04
0.4	0.6	21.99	0.9	0.6	20.41
0.5	0.7	24.65	0.1	0.8	28.96
0.6	0.8	27.30	0.2	0.9	31.62
0.7	0.9	29.96	0.3	0.1	9.59
0.8	0.1	9.16	0.4	0.2	11.56
0.9	0.2	10.33	0.5	0.3	13.59
0.1	0.4	17.44	0.6	0.4	15.96
0.2	0.5	19.81	0.7	0.5	18.33
0.3	0.6	22.33	0.8	0.6	20.7
0.4	0.7	24.98	0.9	0.7	23.32
0.5	0.8	27.63	0.1	0.9	31.95
0.6	0.9	30.29	0.2	0.1	9.84
0.7	0.1	9.14	0.3	0.2	11.80
0.8	0.2	10.58	0.4	0.3	13.89
0.9	0.3	12.54	0.5	0.4	16.26
0.1	0.5	20.11	0.6	0.5	18.63
0.2	0.6	22.66	0.7	0.6	21.00
0.3	0.7	25.31	0.8	0.7	23.65
0.4	0.8	27.97	0.9	0.8	26.31
0.5	0.9	30.62	0.9	0.9	29.29

Table 1 shows the mean percentage errors for different values of α and γ. From the table, we can observe that minimum mean percentage error of 9.12 is obtained for the value α = 0.6 and γ = 0.1. These values are considered for future application predictions during our experimentations.

VM Provisioning:

If we know the nature of jobs i.e., the type of workload we are having, it would be easy and beneficial to assign the jobs to the specific clusters as well as to maintain the requisite amount of resources to them. Thus, once the prediction is done, the applications need to be assigned to the specific clusters based on their nature as whether they are homogeneous or heterogeneous. Same kinds of applications are assigned to the specific clusters suiting their requirements. In our experimentation, the number of clusters in a datacenter is assumed to be 3 by considering three main classes of applications to be executed from the cloud. The application assignments based on the predictions are done as depicted in the algorithm/procedure below.

Procedure:

A = {{No. of Cores, CPU Speed}, RAM, DISK}

C = $\{c_1, c_2, c_3\}$ be the number of clusters

H = $\{h1, h_2, h_3, h_n\}$ be the number of servers/hosts

V = $\{v_1, v_2, v_3, v\}$ be the number of VMs

A = $\{a_1, a_2, a_3, a_n\}$ be the number of jobs/applications

N = Number of available resources

```
Initialize Cthreshold, Rthreshold, Dthreshold, minCthreshold, minRthreshold, minDthreshold
Condition: $C < H \leq V$
$i \leftarrow 1$, count $\leftarrow 0$
While $i \leq n$ do
If(($a_i$≤Cthreshold&&$a_i$>Avgthreshold)&&($a_i$>minRthreshold&&$a_i$ <Avgthreshold)&& ($a_i \geq$ minDthreshold)
$c_1 \leftarrow a_i$
$i \leftarrow i + 1$
Else if (($a_i$>minCthreshold&&$a_i$<Avgthreshold) && ($a_i \leq$ Rthreshold&&$a_i$>Avgthreshold) && ($a_i \geq$ minDthreshold))
$c_2 \leftarrow a_i$
$i \leftarrow i + 1$
Else if (($a_i$<Avgthreshold&&$a_i$≥Avgthreshold) && ($a_i \leq$ Avgthreshold) && ($a_i$>Avgthreshold&&$a_i$≤ Dthreshold))
$c_3 \leftarrow a_i$
$i \leftarrow i + 1$
Else
$c1 \leftarrow a_i$ || $c2 \leftarrow a_i$ || $c3 \leftarrow a_i$
EndIf
If ($a_i \notin$ c1|| $a_i \notin$ c2 || $a_i \notin$ c3)
Add resources to respective clusters
EndIf
EndWhile
End
```

It has been found out from the experiments conducted by the authors [21] that Cthreshold would be 0.8* MIPS capacity and for Rthreshold (0.7 * Memory) and Dthreshold (0.6*Disk capacity). The minThrehsold is set as 0.2* capacity for all the 3 types of resources.

Once the applications have been assigned to a cluster, the selection of virtual machine for the execution needs to be carried out. This job is done by considering the energy consumption and cost factors to be minimal. Although reducing the numbers of active servers may result in reduction of the energy consumption, it may not be true always. For ex: Consider a situation where in an application has been assigned for execution in the cloud datacenter which requires two VMs for its processing.

If there is no mechanism to predict the application type, it would have been assigned to any host available inside the datacenter by considering only the utilization factor and limiting the number of active servers. Thus, it may be possible that the VMs of the same application are assigned to a server of different type which consumes more power compared to assigning VMs on two different servers whose power consumptions as a whole are lesser than assigned to a single sever although all other requirements are fulfilled. This would result in performance degradation, more power consumption and hence higher cost. Thus, it becomes essential to select the best server to place the VMs in it for effective execution of the applications. This procedure results in best assignment since the applications are rightly assigned with their requirements to the appropriate servers consuming minimum energy.

Scenario 1: If the Application is Video Streaming, its requirement is more CPU speed and sufficient RAM capacity with bandwidth.

If it is assigned to a VM which has lesser CPU speed, it results in slower performance and also more VMs may be required to process the requests. Rather we could go for a specific cluster with all VMs having enough capability, then it would result in optimum performance.

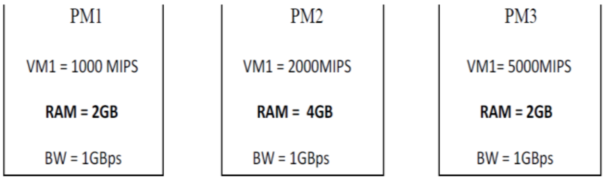

Fig. 2: Application Scenario

In the figure2, if an application request is for video streaming, then the best suit for this would be PM3 and its corresponding virtual machines since their processing speed is high. On contrary to this, if it had been assigned to virtual machines from PM1 or PM2, they would result in performance degradation which in turn leads to more energy consumption and hence the cost. This is because; insufficient speed in processing the applications of this type will result in lower clarity.

Scenario 2: If an application request is for social network or any other applications similar to it, the best suit would be the VMs with higher DISK and RAM capacity. Thus, it would be very essential to separate the servers based on their capacity so that the application requested could be predicted and assigned to the right server and its corresponding virtual machines.

VM provisioning is performed using genetic algorithm, which selects the best virtual machine for placement through a variety of iterations.

The initial job request and the virtual machines for an example scenario is shown in table 2.

Table 2: Initial Job Scenario

Cloudlet ID	MIPS	RAM	Length	VM ID	MIPS	RAM
0	2000	2048	10000	1	2000	2048
1	3000	2048	11000	2	3000	2048
2	2000	2048	12000	3	1000	1024
3	3000	2048	13000	4	1500	1024
4	1500	1024	9000	5	100	256
5	1000	1024	8000	6	500	256
6	500	512	3000	7	4000	2048
7	700	256	2000	8	4000	2048
8	2000	4096	14000	9	1400	512
9	600	512	4000	10	1000	512
10	1000	1536	7000	11	800	512
11	1000	1024	6000	12	700	256

The mapping between the VMs and the servers to the corresponding clusters are shown in the table3. It has been considered to be having 3 clusters supporting 3 different types of applications.

Table 3: Mapping Between the Hosts and the Clusters

Server ID	VM ID	ClusterID
1	1,11	1
2	2,12	1
3	3,13	1
4	4,14	2
5	5,15	2
6	6,16	2
7	7,17	3
8	8.18	3
9	9,19	3
10	10,20	1

The mapping of the cloudlets or the application types for their execution for the given scenario has been shown in table 4 and5. It shows the performance variations based on the changes in the order of assignment to be carried out between the cloudlets and the VMs.

Table 4: Mapping without Order Change

Cloudlet ID	VM ID	Application Type	Time
7	8	A3	0.5
6	7	A3	0.75
1	2	A1	3.67
9	10	A3	4
0	1	A1	5
11	12	A2	8.57
3	4	A1	8.68
10	11	A2	8.79
8	9	A1	10
2	3	A1	12
5	6	A2	16
4	5	A2	90

Table 5: Mapping with Change in Order

Cloudlet ID	VM ID	Application Type	Time
7	11	A3	2.5
1	2	A1	3.67
0	1	A1	5
4	9	A2	6.43
5	10	A2	8
3	4	A1	8.67
8	1	A1	7
2	3	A1	12
11	10	A2	6
10	3	A2	7
6	5	A3	30
9	5	A3	40

It can be seen from the tables 4 and 5, the response time for the task execution is less with change in order compared with no order change strategy.

Energy Model:

It is based on the resource utilization factor of a physical machine and is calculated using the normalized Euclidean distance as utilization vector.

$$D_f = \sqrt{\sum_{i=1}^{n}(|Ui - Ubest|)^2} \tag{7}$$

Here i will be CPU/RAM/Disk. i.e.,

$$D_f = \sqrt{(|U_{ci} - U_{cbest}|)^2 + (|U_{mi} - U_{mbest}|)^2 + (|U_{di} - U_{dbest}|)^2} \tag{8}$$

U_{ci} is the current utilization of the cpu
U_{cbest} is the best utilization of the cpu
U_{mi} is the current utilization of the memory

U_{mbest} is the best utilization of the memory
U_{di} is the current utilization of the disk
U_{dbest} is the best utilization of the disk
U_{max} and U_{min} are the appropriate maximum and minimum values of the utilization which may be considered as 0.8 and 0.2.
The initial best utilization is generally [19] considered as 0.7 and 0.5 for CPU and DISK. It is considered as 0.6 for RAM.
The Overall system utilization factor at any time t [12] is given by,

$$U_{overall} = \Sigma_{i=1}^{N} D_f \quad (9)$$

$U_{overall}$ gives the summation of all the utilization factors ofN systems at any time t.
The overall Energy Efficiency for time T is given by,

$$E = \Sigma_{t=0}^{T} \mathbf{Uoverall} \quad (10)$$

In any cloud provisioning system, VM migration is an important phenomenon which if not handled properly may lead to drastic reduction in performance, increased SLA violation as well as higher cost of ownership for the providers. It could be handled in variety of ways as per the users preference in order to overcome the failures of fulfilling the stated guarantees from the providers. The main 3 steps to decide on VM migration are: identifying when to migrate time, selecting VM for migration and the target host identification.
Genetic algorithm (GA) based optimization is used for VM reallocation in order to reduce the energy efficiency. The selection of the VMs for migration is carried out using a Multi-resource optimization based double threshold policy.

Multi-resource optimization based threshold policy:
This policy considers three status for all the resources considered for the job execution. They are normal, underload and overload.
The normal operation is when processor is utilized above the preset minimum threshold and less than the preset maximum threshold, similarly for the RAM and disk.
The under utilization is the utilization of all the resources being less than 0.2.The over utilization is preset as 0.8 for processor and 0.7 and 0.6 respectively for RAM and disk.

Consolidation Algorithm using GA:
This algorithm has many steps to create the best selection of VMs to be migrated. They are Population initialization, Evaluating the fitness of selection, updating the candidates and checking the candidates in new position.

Population Initialization:
Prepare the tasks/cloudlets to be submitted for execution.
Generate N initial VM requests. N = {n1, n2, n3,.....,np}randomly.
For each request assign a PM based on first fit algorithm PM = (pm1, pm2,pm3,...pmm}
Generate N distribution plans and hence N number of candidates as initial population.
Define the position vector of the candidates as $P^r_s = (p^r_{s1}, p^r_{s2},...,p^r_{sw})$
Where s is the s^th possible solution s<=N,{1, 2,..., w} VM serial numbers(VMIDs)
r is the iteration number.
Thus, the updation of the candidates and the position vector of candidates will lead to a matrix X with values (0, 1).

$$X = \begin{pmatrix} x^r_{s1} \dots & x^r_{sh} \dots & x^r_{sn} \\ x^r_{j1} \dots & x^r_{jh} \dots & x^r_{jn} \\ x^r_{w1} \dots & x^r_{wh} \dots & xr_{wn} \end{pmatrix}$$

If VM j is allocated to node h then X^r_{jh} = 1elseX^r_{jh} = 0.
Since a VM can be allocated to only one PM, we have

$$\Sigma_{h=1}^{n} \quad X^r_{jh} = 1, \forall \ j \in \{1,2,....,m\} \quad (11)$$

Generate Fitness Function using the utilization factor to result in minimized energy consumption as,

$$f(U_{overall}) = \Sigma D_f$$

Here $f(U_{overall})$ is the total energy efficient factors of all the active physical nodes at time 't' after the VMs have been reallocated in the migration queue. In order to fulfill the objective of minimizing energy consumption, fitness function value must be minimized to a lower value. The value of this fitness function determines the best selection of server for execution of the tasks.
Fitness of the cost could be calculated based on the utilization of RAM, DISK and MIPS value of the VMs/host and also the energy consumption to be minimum. This value of the utilization must be considered as a factor for cost fitness. i.e., M is given by,

$$U_{best} - U_i / U_{best}\text{-}U_{worst} \quad (12)$$

The total fitness is given by,

$$F(U_{overall}) + M \quad (13)$$

Update the position of the candidates:
The criteria to be met by the new position of the candidates are:
$x^j_h = \{0, 1\}$ indicates that each VM can be assigned to only one physical node.
$_h\Sigma x^j_h = 1$□ j Here, x^j_h indicates whether the VM j is assigned to node h or not.
If VM j is assigned to host h then, $x^j_h = 1$ else $x^j_h = 0$

$$({}_j\Sigma_r{}^{CPU}{}_j * x^j_h \leq c^{CPU}{}_h) \wedge ({}_j\Sigma_r{}^{DISK}{}_j * x^j_h \leq c^{DISK}{}_h) \wedge ({}_j\Sigma_r{}^{RAM}{}_j * x^j_h \leq c^{RAM}{}_h) \quad (14)$$

Once all the constraints are satisfied, candidates will be updated to the new positions[12], else original value is retained.

a) If $\Sigma^n_{h=1} s^r_{jh} > 1$, VM j is allocated to multiple physical nodes. Then set □h, $s^j_{hr} = 0$ and sort all physical nodes in the ascending order. The VM j is allocated to the first physical node which satisfies the above formula. Otherwise candidates will not be updated.
b) If $\Sigma^n_{h=1} s^r_{jh} = 1$, VM j is allocated to a physical node, then check whether constraint is satisfied. If it is satisfied, the candidates are updated to new positions else, they are not updated.
c) If $\Sigma^n_{h=1} s^r_{jh} = 0$, VM j is not assigned to any physical node. The new positions do not satisfy the conditions, hence candidates are not updated.

This VM migration and consolidation algorithm is based on multi-resource energy efficient model using Genetic algorithm. The energy efficiency and SLA violation are mainly considered based on migration of VMs which are most likely to cause SLA violation. First fit algorithm is used to generate candidates for reducing active physial nodes and a Genetic Optimization Algorithm is introduced and designed to updatepositions of the candidates. Normalized Euclidean distance is defined to evaluate energy efficiency after migration.
In the proposed algorithm, m VMs and n physical machines are assumed in the cloud datacenter and v VMs in migration queue.

4. Results & analysis

The proposed mechanism is compared with the existing system which does not use any application prediction and cluster creation. The planet lab workload from the CoMoN project has been considered for the experimentation.
Initially the comparison is done for the energy consumption across the number of servers in the cluster. It could be inferred from the figure 4 that the proposed algorithm results in upto 51.19% lower energy consumption compared to the existing algorithms.

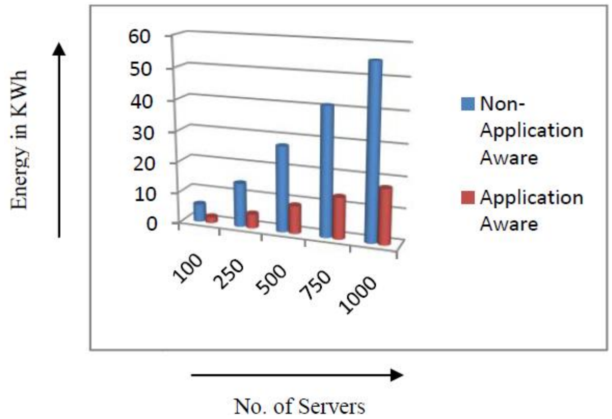

Fig. 4: Energy Consumption vs. No of servers.

Figure 5 depicts the variation with respect to the number of VMs and the number of migration that has been carried out to successfully execute the tasks.

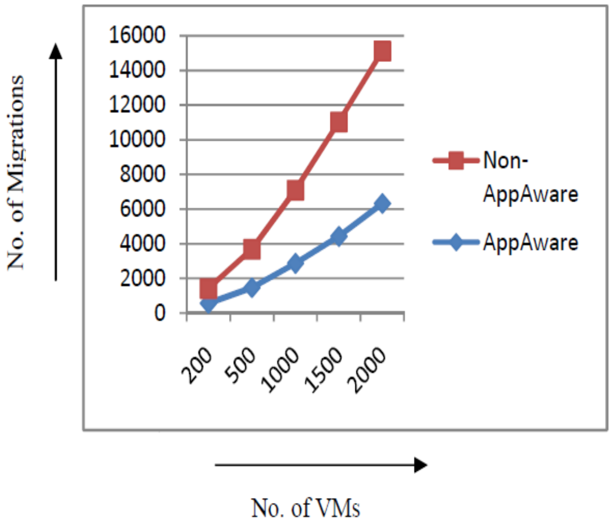

Fig. 5: VM Migrations vs. No. of VMs.

Based on the data from the figure 5 it can be understood that no. of migrations are more in an environment with no applications compared with the proposed mechanisms. It is found to be resulting in 19.16% lesser migrations compared to the existing algorithms.

The proposed mechanism has also been compared for the SLA violations during processing of the tasks. The figure 6 presents clearly that the SLA violations are 25.95% lesser compared to the existing non-application aware mechanisms.

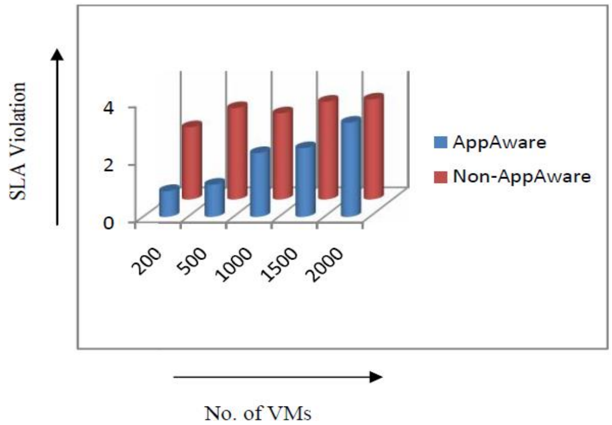

Fig. 6: SLA Violations vs No. of VMs.

The Resource utilization of the proposed system is better than the existing mechanism as shown in the figure 7. It has been estimated to be 10.45% better than it.

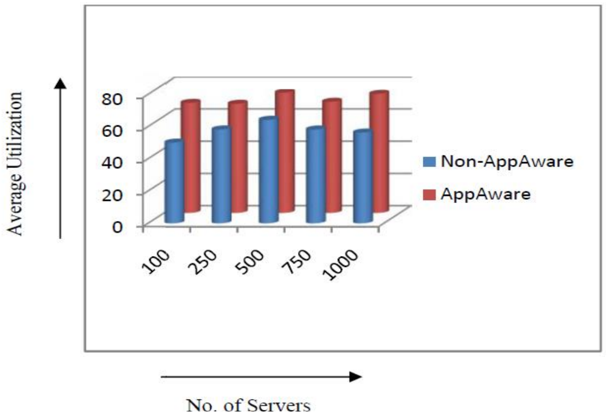

Fig. 7: Avg. Utilization vs. No. of Servers

The mean total execution time for the tasks is shown in the figure 8. The number of virtual machines and the corresponding execution time in seconds clearly indicates that the proposed methodology results in about 46.18% reduced time.

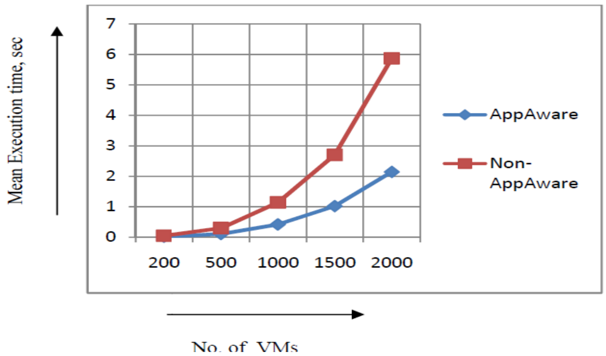

Fig. 8: Mean Execution Time vs. No. of VMs.

The number of active servers specifically contributing for the execution of the tasks are shown in figure 9. It is clear that the average number of the active servers and the corresponding number of virtual machines available for the job s are very less in the proposed system thereby leading to better consolidation of the servers.

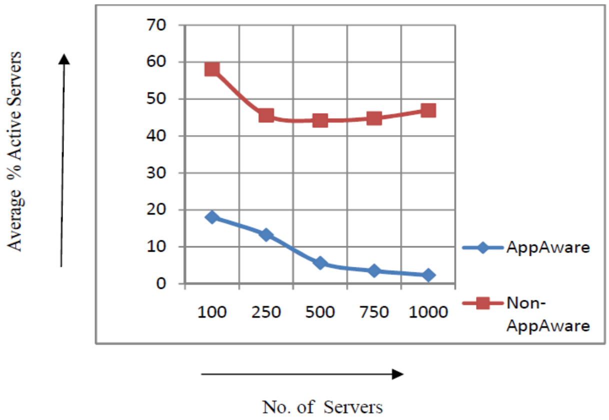

Fig. 9: Avg. Active Servers vs. No. of VMS.

It is also found that the average rate of load balancing across the servers is 21.52% better compared with the existing algorithm.

5. Conclusion

In the cloud environment, many factors need to be considered during resource provisioning. Among them the muchneeded factor to achieve the target benefits for the providers is cutting down the cost of resource provisioning. This requires rigorous measures to optimize the usage of the different resources thereby enhancing the efficiency in all respects. The proposed mechanism shows improved effectiveness in resource allocation compared with existing algorithms. The results obtained and its validation after the experiments with no. of application types as 3 using CloudSim toolkit shows that the proposed scheme has resulted in significant reduction (savings) in energy consumption, no. of migrations, SLA violations, execution time, no. of active servers andbetter utilization. It is found that when the types of applications are other than 3 also there is no significant reduction in the obtained benefits.

The future scope could be real time implementation using some specific workloads and considering some other factors like bandwidth also for validating the efficiency of the proposed approach.

References

[1] Wanneng Shu, Wei Wang and Yunji Wang," A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing"EURASIP Journal on Wireless Communications and Networking, 2014:64, Springer.https://doi.org/10.1186/1687-1499-2014-64.

[2] Rongdong Hu, Jingfei Jiang, Guangming Liu and Lixin Wang, "Efficient Resources Provisioning Based on Load Lo Forecasting in Cloud" The Scientific World Journal Hindawi Publishing Coorporation, Volume 2014, Article 321231, 12 pages ID https://doi.org/10.1155/2014/321231.

[3] MyintMyatMyo and Thandar Thein, "Efficient Resource Allocation for Green Data Center", Proceedings of the third International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2014) Feb. 11-12, 2014, Singapore.

[4] Wanneng Shu, Wei Wang and Yunji Wang "A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing" EURASIP Journal on Wireless Communications and Networking 2014, 2014:64, Springer

[5] Rongdong Hu, Jingfei Jiang, Guangming Liu and Lixin Wang, "Efficient Resources Provisioning Based on Load Forecasting in Cloud", The Scientific World Journal Hindawi Publishing Coorporation, Article ID 321231, 12 pages, Volume 2014.

[6] Adel Nadjaran Toosi and Rajkumar Buyya, "Fuzzy Logic-based Controller for Cost and Energy Efficient Load Balancing in Geo-Distributed Data Centers", Proceedings of 8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC), 2015, pp. 186-194,ISBN: 978-0-7695-5697-0, DOI 10.1109/UCC.2015.35

[7] R. R. Darwish, "Autonomic Power Aware Cloud Re source Orchestration Architecture for Web Applications", International Journal of Grid and Distributed Computing SERSC publications, Vol. 6, No. 6, pp. 63-82, 2013.

[8] Sanket Dangi, Deepthi Karnam, Celina Madhavan, Sudha Mani and Shrisha Rao", Self-tuning Energy-Aware Ensemble Model for Server Clusters", Annual International Conference on Green Information Technology – Green IT 2010, ISBN: 978-981-08-7240-3, 2010, doi: 10.5176/978-981-08-7240-3 G-33.

[9] Anton Beloglazov and Rajkumar Buyya, "Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers", MGC 2010, ISBN: 978-1-4503-0453-5/10/11, ACM 2010 https://doi.org/10.1145/1890799.1890803.

[10] Sukhpal Singh and Inderveer Chana, "Energy based Efficient Resource Scheduling: A Step Towards Green Computing", International Journal of Energy, Information and Communications (IJEIC) SERSC Vol.5, Issue 2, 2014.

[11] Rodrigo N. Calheiros, Rajiv Ranjan and Rajkumar Buyya, "Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments", Proceedings of IEEE International Conference on Parallel Processing, 2011.

[12] Hongjian Li, Guofeng Zhu, Chengyuan Cui, Hong Tang, Yusheng Dou and Chen He, "Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing", Computing (2016) 98:303–317 DOI 10.1007/s00607-015-0467-4, Springer-Verlag Wien, 2015.

[13] Ali Al-maamari and Fatma A. Omara," Task Scheduling Using PSO Algorithm in Cloud Computing Environments", International Journal of Grid Distribution Computing Vol. 8, No.5, (2015), pp.245-256, ISSN:2005-4262, IJGDC.

[14] Elina Pacini and Cristian Mateos," Dynamic Scheduling based on Particle Swarm Optimization for Cloud-based Scientific Experiments", CLEI ELECTRONIC JOURNAL, VOLUME 14, NUMBER 1, 2014.

[15] Madhukar Shelar, Shirish Sane and Vilas Kharat, "Enhancing Performance of Applications in Cloud using Hybrid Scaling Technique" International Journal of Computer Applications (0975 – 8887), Volume 143, No.2, 2016.

[16] Shreenath Acharya and Demian Antony D'Mello, "A Taxonomy of Live Virtual Machine (VM) Migration Mechanisms in Cloud Computing Environment", Proceedings of the IEEE International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), 2013, pp. 809-815.

[17] Raghavendra Achar, Santhi Thilagam, "Application nature aware virtual machine provisioning in cloud", International Journal of Ad Hoc and Ubiquitous Computing", Vol. 27, Issue 2, Inderscience Publishers, 2018.

[18] Shreenath Acharya, Demian Antony D'Mello, "Enhanced Dynamic Load Balancing Algorithm of Resource Provisioning in Cloud", Proceedings of IEEE International Conference on Inventive Computation Technologies (ICICT), 2016.

[19] A.I.Awada, N.A.El-Hefnawyb and H.M.Abdel_kader, "Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments", International Conference on Communication, Management and Information Technology (ICCMIT 2015)", Procedia Computer Science 65 (2015) 920 – 929, Elsevier. https://doi.org/10.1016/j.procs.2015.09.064.

[20] Ankita Atrey, Nikita Jain and Iyengar N.Ch.S.N, "A Study on Green Cloud Computing", International Journal of Grid and Distributed Computing Vol.6, No.6 (2013), pp.93-102.https://doi.org/10.14257/ijgdc.2013.6.6.08.

[21] Srikantaiah S, Kansal A, and Zhao F, "Energy aware consolidation for cloud computing", Proceedings of conference on Power aware computing and systems, p. 1-5, USENIX Association Berkeley, 2008.

[22] Xu Yi-Chun, Xiao Ren-Bin, "An improved binary particle swarm optimizer", Pattern Recognition Artificial Intelligence, 20(6), 788-793, 2007.

[23] Darrel Whitley, "A genetic algorithm tutorial", Statistics and Computing, Vol. 4, Issue 2, pp. 65-85, Springer, 1994.

[24] https://scm.ncsu.edu/scm-articles/article/double exponential-smoothing-approaches-to-forecasting-a-tutorial visited 18 August 2018.