A Dynamic Energy efficient Resource Allocation Scheme for Heterogeneous clouds using Bin-Packing Heuristics

Ragini Karwayun
Researcher
Mewar University, Rajasthan, India

Dr. K. P. Yadav
Vice Chancellor
Sangam University, Bhilwara , Rajasthan, India

Abstract- One of the main challenges of cloud computing is to assign workloads to the hosts available in the data centers with the help of virtualization, i.e., by using virtual machines. Therefore, suitable resource allocation and load balancing techniques are required for implementing efficient cloud environments. The users task need VM allocations for completing the computation work. Moreover, VM's need to be submitted to the physical hosts in a cloud data center.

The allocation of resources is independent of execution priorities of the user's task. Popular resource allocation algorithms such as FCFS, SJF and Round Robin are greedy in nature and perform unfair allocation. In this research, we have designed a different dynamic resource allocation scheme for cloud environment. It is modelled on bin packing approach where each VM needs to be hosted on physical servers. As the user's tasks arrive in the system, the workload is represented as cloudlets and VM are used to execute these cloudlets. As the use of cloud computing is scaling new heights, energy consumption optimization is gaining primary focus in the implementation of cloud data centers.

The goal is to use minimum number of bins with each task having a fair share of requested resources along with energy consumption calculator keeping track of over loaded and under loaded hosts, so that proper load balancing can be achieved to save energy consumption. The proposed fair resource allocation scheme is based on first fit approximation heuristic with load balancing features for efficient power consumption.

Keywords – Resource Allocation, Energy Efficient, Clouds, Approximation.

I. INTRODUCTION

The objective of a good resource allocation scheme is to allocate the incoming user's jobs to virtual resources in the cloud servers, and each server has a limited capacity. The most important goal is to allocate appropriate resources to the jobs in order to achieve well balanced load across all virtual servers, and in addition, keeping the number of physical servers as less as possible.

The IaaS model for cloud service providers can be conceptualized as a form of in line bin packing problem, where bins symbolizes physical resources and items represent virtual machines having dynamic loads. The input provided to the problem is a sequence of operations, each involving an insertion, deletion or updating the item size of an item.

The objective is to have packings with a small number of active bins. To have less numbers of bins is important to save energy.

Many researchers have proposed various schemes for having nearly optimal resource allocation scheme with a good task scheduling and load balancing algorithms. They have used different bin packing algorithms like Best Fit, Next Fit and First Fit, to achieve the desired objective. Some have used the offline while others have proposed on line bin packing scheme. The various proposed algorithms are modified versions of the existing bin packing algorithms. [6,7,8]

1.1 Bin Packing

Bin packing is a classical problem of optimization defined as follows:

"Given a set of items, the objective is to pack the items into a minimum number of bins such that the total size of the items in each bin does not exceed the bin capacity and any item cannot occupy more than one bin." It can be

formulated as follows: Given a sequence of items ={1,2,...,n} with sizes s1,s2,...,sn ∈ (0,1] drawn from a uniform distribution, find an allocation of the items into sets of size 1 (called bins) so that the number of bins used is minimized [1]. Moreover, the sum of the sizes of the items assigned to any bin should not exceed its capacity. A bin is empty if it contains no item, else it is occupied.

In the classical bin packing problem, we want to pack a sequence of items, each with size in the range [0, 1] into a minimum number of unit-size bins. Dynamic bin packing is a generalization of the classical bin packing problem introduced in [2]. This generalization assumes that items may depart at arbitrary time. The objective is to minimize the maximum number of bins ever used over all time.

1.2 Bin Packing Algorithms

The bin packing (BP) problem is a minimization optimization problem defined as follows.

"Given a list of objects and their weights, and a collection of bins of fixed size, find the minimum number of bins so that all of the objects are assigned to a bin. Few of the algorithms are described in the following discussion.

1.2.1Approximation On Line Algorithms

Next Fit:

Next Fit considers the most recent partially filled bin for allocation. It has only one active bin into which it packs the incoming item. When the free space in this bin becomes too small to accommodate the next item, a new active bin is opened and it never uses the previous again.

First Fit:

First Fit achieves better performance in terms of number of used bins but a worse running time as it considers all non-empty bins active and tries to pack every item in these bins before opening a new one. If it can find no active bin to accommodate the new request, it opens a new bin and allocate the item in the new bin. So, the restriction of using a single bin is removed entirely and all partially filled bins are considered as possible destinations for the item to be allocated.

Best Fit:

Best Fit is the best-known algorithm for on-line bin packing and has the best worst case and average uniform case. Best Fit (BF) picks (among the possible bins for the item) the one where the amount of free space is least. It picks the bin with the least amount of free space which can still accommodate the current item.

1.2.2 Offline Algorithms

An offline algorithm simply repacks everything each time an item arrives. Packing large items is difficult with an online algorithm, especially when such items occur later in the sequence. There are two important offline algorithms for bin packing. They are First Fit Decreasing and Best Fit Decreasing.

First Fit Decreasing

This algorithm first sorts items in non-increasing order with respect to their size and then processes items as the First Fit algorithm.

Best Fit Decreasing

This algorithm also initially sorts items in non-increasing order with respect to their size and then processes them sequentially. It differs from the previous algorithm in the selection of the bin for inserting the new item. Best Fit Decreasing chooses a bin which is best suited for the item, least empty space is left after the item is packed into a bin.

1.2.3. Online Algorithms

Harmonic Algorithm

The algorithms define types for the items based on their sizes and for each type the items are packed in dedicated bins which contain only items of one specific type. Items of one type releases at a time, then departs most of them from the packing, without emptying any bin, and proceeds to release items of a different type. Previously opened bins will not pack items of a different type, thus the wasted space in existing bins is maximized.

II. PROPOSED RESOURCE ALLOCATION ALGORITHM

2.1 Dynamic resource allocation problem consists of three basic components:

a. Resource allocation
b. Task scheduling
c. Load Balancing.

Energy consumption of IT infrastructure is very high. The reason behind this high value is the inefficient use of computing resources. In a research[3] data collected from a large number of computing servers show that most of the time servers perform at approximately 15 % of their maximum capacity.
This increases the total cost of acquisition due to over provisioning. It has also been found that idle servers also consume about 70% of the maximum power consumption at high load. [4]

Using virtualization and live migration of VMs, one can improve resource utilization by creating multiple VMs on a physical server. By continuously monitoring the load on the servers we can identify the overloaded and under-loaded hosts. After selecting VMs for migration from overloaded and under-loaded hosts, we can have effective resource utilization and a balanced power consumption.

Due to the highly fluctuating workloads in today's service applications in clouds, it is very complex procedure to strike a balance between consolidation of servers and maintaining SLA parameters. Excessive high migration can result in performance degradation, thus violating the Qos defined in SLA. Therefore, the resource allocation algorithm need to strike a trade-off between these two parameters. [15,16]

We have tried to propose a scheme that tries to use the positive features of various discussed algorithms in order to define an energy efficient resource allocation scheme.

2.2 System Model

Model considers IaaS environment as the target system, where providers are allocating physical resources instances to user's application through virtualisation. Providers pack user's applications and requirements of resource requirements into virtual machines. The system is represented by a huge data center. In conventional Bin packing VM policy, hosts are considered as Bins, which need to be filled up with Items (VMs).

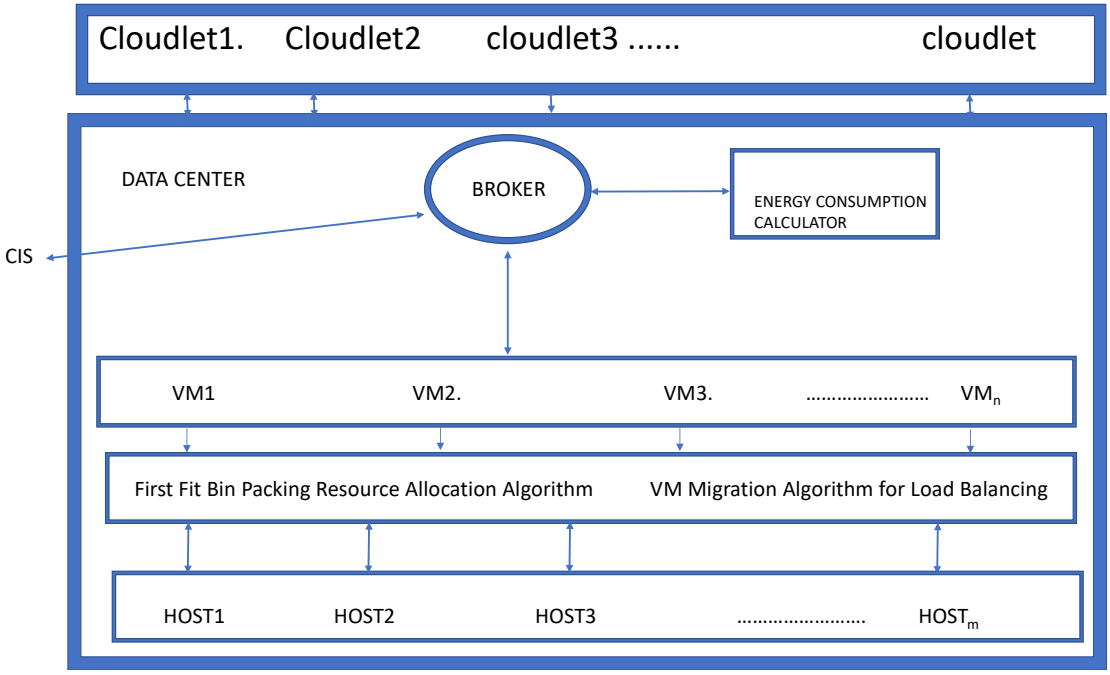

Fig. 2.1 System Architecture

The proposed system has three major components: -

1. FF Bin Packing module for allocating VMs to the hosts. It's an online dynamic mapping algorithm, that is modelled on conventional Bin packing approximation algorithm. As the Bin packing problem is NP-Hard, we can use a heuristic for the approximation version. The algorithm allocates the VMs to hosts in the data center as they arrive in the system. It uses First fit heuristic which is almost as good as Best fit approach but don't need the sorting of bins on its parameters.
2. VM Migration Scheduler for balancing the load in the system as the VMs leave the system after completion. The data center Broker uses Energy Consumption Calculator to identify the overloaded and under-loaded physical nodes. It then selects the source server, VM for migration and the sink server, where this migrated VM will be mapped.
3. Energy Consumption monitor – identifies the source and the sink server for VM migration along with the VM for migration. It uses two thresholds – upper and lower threshold values to decide overloaded and under-loaded hosts.

Lightly loaded servers are identified that have their power consumption below the lower threshold to aid in green computing. Its VM are migrated to other hosts so that these servers can be switched off to save energy. Many researchers have shown that energy consumption of idle servers is approximately around 70% of its maximum power consumption on full load. However, few idle servers need to be kept in sleep mode so that they can be used in case of sudden peaks in workloads.

To describe the system model, we define following elements :-

1. There are m multi capacity Bins : Bin_1, Bin_2,……Bin_m.
2. Each bin is characterized by few parameters viz., CPU, RAM, PE(processing elements), Bandwidth, Storage and current Power consumption. $CP_i = \{CP_{i,1}, CP_{i,2}, ….. CP_{i,5}\}$ where $CP_{i,2}$ is the 2^{nd} parameter(RAM) of it i^{th} host.
3. There are n variable size items (IT_1,IT_2,….IT_n) in the set ITEM. These n items need to be packed in as few bins as possible without overstepping on bin's capacity.
4. An item ITj in the set ITEM is represented as $IT_j = \{IT_{j,1}, IT_{j,2}, ….. IT_{j,5}\}$ where $IT_{j,2}$ is the 2^{nd} parameter(RAM) of it j^{th} item.
5. The 6 parameters are: CPU(MIPS), RAM, PE, Bandwidth, Storage and Power consumption.
6. We need to define a mapping function for items to be mapped to bins such that all items are successfully mapped to bins.

   $\text{function}(j) \rightarrow i \quad \forall\ i = 1 \text{ to } m, \forall\ j = 1 \text{ to } n$

   $\forall\ j\ \text{function}(j) \neq \varnothing$ ..……..…(A)

   $\forall\ p = 1….6 \quad \Sigma_{j=1..n}\, a_{j,i} IT_{j,p} < CP_{i,p}$ …….(B)

   where $a_{j,i}$ is a Boolean variable such that $a_{j,i} = 1$ it item j is assigned to bin i and $a_{j,I} = 0$ otherwise. Equation B states that the sum of requirement for a parameter of all the items mapped to a bin should be less that the capacity of the bin. That is, if two items mapped to a bin requires 250 MB of RAM each, than the respective bin should have more than 500 MB of RAM.
7. Let us assume that the maximum power capacity of each host is same = Powermax.
8. A decision variable h_i is defined for each server such that

   $h_i = 1$ if server i is hosting some VMs

   $h_i = 0$ if server i is idle

The optimized allocation algorithm is subject to following constraints:

i. Each host cannot exceed its maximum power limit.

$$\Sigma_{j=1..n}\, IT_{j,6}\, a_{ji} \leq \text{Powermax} - Cp_{i,6} \quad \forall\ i = 1 \text{ to } m \qquad \text{……(C)}$$

ii. As per the SLA, each VM should be assigned to one and only one server.

$$\Sigma_{i=1..m}\, a_{ji} = 1 \;\; \forall\, j = 1 \text{ to } n \qquad \text{……..(D)}$$

The VM migration algorithm is subject to following constraints:

i. The power utilization of each host should be less that equal to upper_threshold value.

ii. The power consumed in migrating a VM should be less than the power saved at the host.

iii. The difference between the total power consumption of the idle servers and the total cost of migration of VMs that are candidate for migration must be as big as possible.

If m is the number of hosts available in data center and m_u is the number of used servers whose power consumption is less than Powermax, where $m_u < m$

$$\text{Powerutil} = \Sigma_{i=1..m}[\text{Poweridle}*h\_idle_i] - [\text{VMmigrationCost}_k * X_{i,jk}] \ldots (E)$$

$$\forall\, j = 1 \text{ to } m \;\; \forall\, i = 1 \text{ to } m \qquad \forall\, k = 1 \text{ to } n$$

where, Poweridle is the power consumption of idle server. Let us assume that it is same for all servers.

VMmigrationCost is the migration cost of VM_j. It can be calculated as a factor of Memory used by the VM by the bandwidth used by it,

$$\text{VMmigratioCost}_k = IT_{k,2} / IT_{k,4}$$

$X_{i,jk} = 1$ if VM_k is migrated from host i to host j

$h\_idle_i = 1$ if host i is idle else it is 0

[6,8] have shown that there is a linear relationship between CPU utilization and power consumption. Let cpuutil be the percentage of CPU utilization $\in \{0,1\}$.

$$P(\text{cpuutil}) = \propto * \text{powermax} + (1-\propto)\text{powermax}*\text{cpuutil for o} \leq \propto \leq 1$$

$$= \text{powermax}(\propto + (1-\propto)\text{cpuutil})$$

As idle servers also consume approximately 70% of the maximum power consumption of the servers. So let $\propto = 0.7$

$$P(\text{cpuutil}) = \text{powermax}(0.7 + 0.3\text{cpuutil})$$

Due to the dynamic nature of the system, once the job is completed, VMs will leave the system, leaving it in an unbalanced state. It may happen that the power consumption of a server drops down below the lower threshold value. Hence, there is a need to have a re-allocation algorithm work that may optimize the resource utilization. The objective of the algorithm is to identify the overloaded and under-loaded hosts, and migrate some VMs, to strike a balance. Selecting the VMs to migrate is a tricky task, as VM migration is an overhead involving some cost.

2.2.1 ALGORITHM 1 –[First Fit Bin Packing Resource Allocation Algorithm]

Input : VMList → List of VMs (Items)

PMList → List of PMs (Bins)

For each ITj of [1..n] in VMList

Select first PM of [1..m] from PMList

{ For k = 1 to 6 //***parameter

IF ITj,k $\leq$ CPi,k

Set aji = 1 //*** VM j is allocated to host i

CPi,k = CPi,k + ITj,k

ELSE

aji = 0 //*** VM j is not allocated to host i

} //** Next PM

} //** Next VM

2.2.2 ALGORITHM 2 – [Energy Consumption Calculator]

Input : VMlist, PMlist

Output: PMoverLoadedList , PMunderLoadedList, VMmigrationList

FOR each PMi find power consumption CPi,6

IF CPi,6 > upper_threshold add PMi to PMoverLoadedList

ELSE IF CPi,6 < lower_threshold add PMi to PMunderLoadedList

ELSE add PMi to PMavialableList

FOR each PMi in PMoverLoadedList

{ Sort all VMs in descending order of VMmigrationCost

FOR each VMj in PMi

IF $IT_{j,6} > CP_{i,6}$ – upper_threshold

{ X = $IT_{j,6}$ – [$CP_{i,6}$ – upper_threshold]

IF X < maxUtil

{ maxUtil = X

add VMj to VMmigrationList

$CP_{i,6}$ = CPi,6 – $It_{j,6}$

}

}

}

FOR each PMi in PMunderLoadedList

FOR each VMj in PMi

Add VMj to VMmigrationList

2.2.3 ALGORITHM 3 [Dynamiv VM Reallocation]

Input : PMavialableList, VMmigrationList

Output: VM to PM allocation

Sort VMmigrationList in decreasing order of ITj,6

FOR each VMj in VMmigrationList

```
{  FOR each PMi in PMavialableList
   { For k = 1 to 6  //***parameter
    IF ITj,k ≤ CPi,k
     {   Set aji = 1 //*** VM j is allocated to host i
        CPi,k = CPi,k + ITj,k
      }
     }
   }
```

Upon arrival of new VM placement and resource requests the algorithms select the most appropriate nodes to host new VMs from the available and active nodes. Whenever necessary these algorithms may resort to turning new nodes on, when the set of active nodes are full and cannot host the new arriving VM instances. Both algorithms will attempt serving the requests in the currently active nodes and will of course typically avoid turning any new nodes on.

The algorithms are combined with the Energy consumption monitor and VM reallocation algorithm that is launched if a number of VM jobs terminate since their dedicated resources become available for opportunistic reuse and for more efficient resource allocation and distribution. These departures are the opportunity for the consolidation algorithm to rearrange allocations by moving VMs into the smallest possible set of nodes. All emptied or freed servers (or nodes) are turned off to minimize energy consumption.

The consolidation is achieved by the VM reallocation algorithm that moves VMs from selected source nodes to selected destination nodes. The end result is the activation and use of the smallest set of nodes in the data centers.

III. EXPERIMENTAL RESULTS

The proposed System for Dynamic load balancing and dynamic resource allocation for Cloud Computing Environment is deployed using CloudSim. CloudSim is a simulation toolkit that provides simulation of cloud computing environments using virtual machines which need to be managed. It provides data center to simulate and deploy the application. It also provides mapping for utilizing the resources. In CloudSim, there are major entities like Data Center, Management Information System, and Cloud broker[15, 16].

The host, VM and cloudlet configurations are specified according to the following table1 and table 2

Paraameter	HOST	VM
CPU	1000,2000,3000 MIPS	250,500,750,1000 MIPS
RAM	10000 MB	128 MB
Bandwidth	100000Mbps	2500 Mbps
Storage	1000 GB	2500 MB
Processing Element	1	1

Table 1

Cloudlet

No. of lines	150000
Processing Element	1
File size	300
Output Size	300

Table 2

We performed the algorithm on five different configurations like:

	HOST	VM	Cloudlet
Configuration 1 :	5	15	30
Configuration 2 :	10	25	50
Configuration 3 :	15	40	80
Configuration 4 :	20	60	120
Configuration 5 :	25	100	200

Table 3

3.1 PM Requirement Comparison Graph

The proposed algorithm aims to minimize the number of hosts allocated by using the first fit approximation algorithm and minimizing the energy consumption of the data center. Initially, the jobs increase linearly in the proposed system. Figure 3.1 gives the performance of our algorithm against the best fit and worst fit approximation algorithms for the above five different configurations.

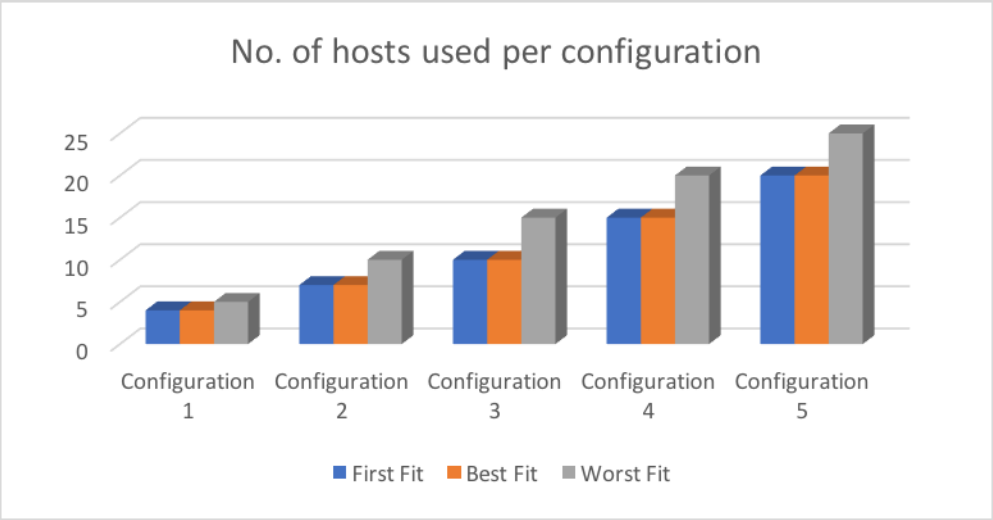

Fig. 3.1 Number of hosts used per configuration

4.2 Energy Consumption.

The following graph gives the energy consumption comparison of our energy efficient resource allocation algorithm against the performance of the algorithm without using the energy calculator module. It is quite clear from the observations that our algorithm takes care of reduced resource allocation as well as maintain a fair load balance in the data center thus helping in green computing. Fig 3.2 gives the energy consumption in Kwh.

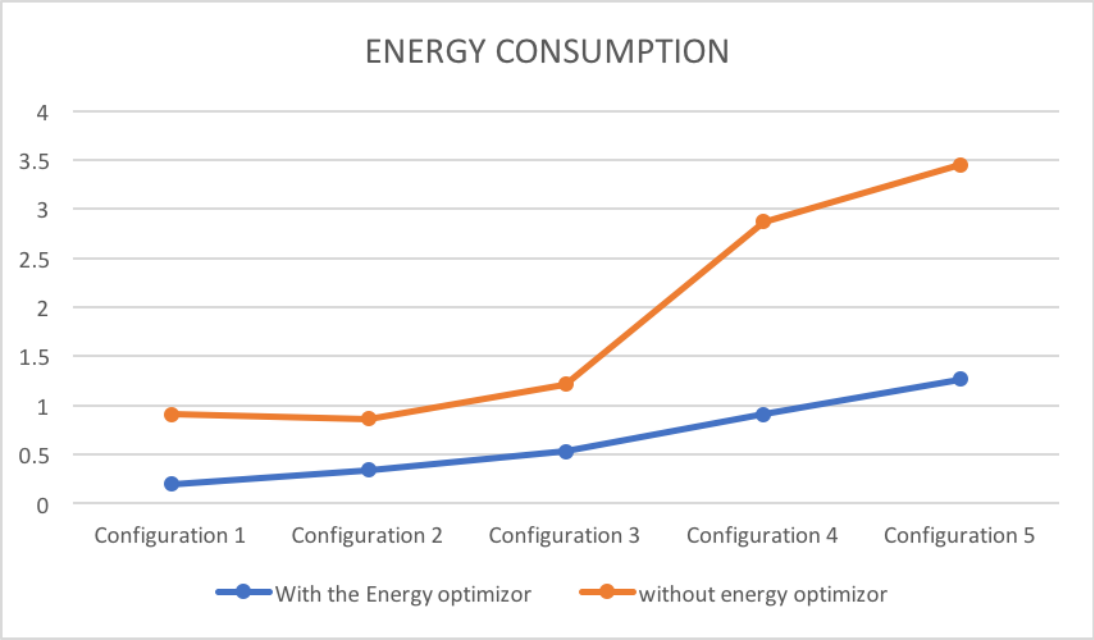

Fig. 3.2Energy consumption with and without host consolidation

IV.CONCLUSION

We have proposed a bin packing based Approach for Energy Efficient Resource Allocation for Classical IaaS clouds. We have formulated the problem of energy efficient resource allocation as a bin-packing model. This model is VM based and provides on-demand resource allocation. We propose an Online dynamic resource algorithm for initial resource allocation. To deal with dynamic resource consolidation, an algorithm for dynamic VM reallocation is also proposed. It is based on VM migration and aims to optimize constantly the energy efficiency after service departures. An energy consumption calculator is used that monitors the load of the system identifying over loaded and under loaded systems and also identifying target machines to share the load. Experimental results show benefits of combining the allocation and migration algorithms and demonstrate their ability to achieve significant energy savings while maintaining feasible runtimes.

Some issues related to the energy efficient resource allocation problem in Cloud environments have not been addressed in this paper, these limitations will be addressed as future work. A major issue in resource allocation in clouds deals with reserving the cloud resources in advance. Many times, it happens that the resources requested by a client application are not currently available, leading to the refusal of the resource request. Advance reservation will ensure the availability of requested resource on time. In addition to this, we can also incorporate a negotiation mechanism for alternate slots if there are more reservation requests than the available slots.

REFERENCES

[1] T. Gautier, X. Besseron, and L. Pigeon, “Kaapi: A thread scheduling runtime system for data flow computations on cluster of multi-processors,” in PASCO ’07. New York, NY, USA: ACM, 2007 .

[2] S. Wang, K. Van, W. Liao, and S. Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICC SIT), Chengdu, China, pp.108-113, September 2010.

[3] Anton Beloglazov and Rajkumar Buyya “Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers” Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia.

[4] Anton Beloglazov and Rajkumar Buyya “Energy Efficient Resource Management in Virtualized Cloud Data Centers”, Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia.

[5] Zhang Hui et al “Intelligent Workload Factoring for A Hybrid Cloud Computing Model” NEC Laboratories America

[6] Yaohui CHANG "Energy Efficient Resource Selection and Allocation Strategy for Virtual Machine Consolidation in Cloud Data centers" IEICE TRANS. INF. & SYST., VOL.E101–D, NO.7 JULY 2018

[7] Yusen Li et al "Dynamic Bin Packing for On-Demand Cloud Resource Allocation" This research is supported by Multi-Platform Game Innovation Centre (MAGIC), funded by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority.

[8] Shahin Kamali "Efficient Bin Packing Algorithms for Resource Provisioning in the Cloud"

[9] ALEXANDER NGENZI et al "DYNAMIC RESOURCE MANAGEMENT IN CLOUD DATA CENTERS FOR SERVER CONSOLIDATION"

[10] Joan Boyar et al "The Maximum Resource Bin Packing Problem", supported in part by the Danish Natural Science Research Council (SNF), and Israel Science Foundation (ISF).

[11] Dilip Kumar et al "Energy Efficient Heuristic Resource Allocation for Cloud Computing" 2014.

[12] Dinesh Komarasamy et al "A Novel Approach for Dynamic Load Balancing with Effective Bin Packing and VM Reconfiguration in Cloud" Indian Journal of Science and Technology, *Vol 9(11), DOI:* 10.17485/ijst/2016/v9i11/89290, March 2016

[13] Arya M B et al "A Combined Bin Packing VM Allocation and Minimum Loaded VM Migration Approach For Load Balancing In IaaS Cloud Data centers" International Journal of Research in Engineering and Technology eISSN: 2319-1163 | pISSN: 2321-7308

[14] Muthaiah U. et al " Dynamic Bin Packing for Resource Allocation in the Cloud data center", IJORAT Vol I Issue 3, March 2016

[15] Madhumati R. et al "Dynamic Resource Provisioning for the Cloud using Bin Packing Technique" IJRSI Volume II, Issue X, October 2015

[16] Solanki N. et al "Energy Aware Virtual Machine Allocations using Bin Packing Algorithms in Cloud Data Centers", IJERT Vol 5 Issue 12, Dec'2016