Dynamic Resource Provisioning in Datacenters using Profitability-aware VM Placement

Narander Kumar and Swati Saxena

Department of Computer Science
Babasaheb Bhimrao Ambedkar University (A Central University)
Lucknow

Abstract: With the ever-spreading user-base in cloud datacenters, it has become increasingly challenging to satisfy each user's service demand. Resource requirements or service demands of cloud users are embedded in virtual machines (VM). Virtual machines are software extensions of physical machines such that a single physical machine can generate a number of virtual machines and hence, can service multiple users. VM placement is the task of allocating a VM to a chosen physical machine (PM) such that the resource demands of that VM are satisfied by the physical machine. An effective VM placement aims at balancing load in the datacenter thereby avoids unnecessary VM migrations and improves the performance of a datacenter. Bin packing has been widely used as a technique for VM placement in the variations of first-fit, best-fit and/or worst-fit. This paper proposes a novel VM placement technique which ensures that the profits to the cloud service provider are maximized, establishes fairness and service availability to the cloud user by dynamically placing a virtual machine to the best available physical machine using the concept of clusters. Our VM placement strategy avoids unnecessary migrations in the datacenter by balancing load on all the physical machines present inside a cluster. A comparison, of the proposed technique with First-Fit bin packing placement technique, highlights a better time efficiency and resources utilization in the present work.

Keywords: Virtual Machine Placement, Resource Utilization, Load Balance, Profit

1. Introduction

Resource distribution/provisioning in a cloud datacenter takes place in the form of VM placement. VM placement involves selection of an appropriate physical machine to host a single or group of virtual machines. It comprises carefully monitoring the resource utilization of the physical machine, its remaining capacity, the demands of an incoming virtual machine, its negotiated Quality of service (QoS) parameters, the overall effect of placing the incoming virtual machine on the performance of the datacenter and the revenues earned by the cloud service provider after a successful placement. It can be an initial placement where all the PMs are idle and incoming VMs are placed on them for the first time. Placement can also be subsequent where an incoming service request (VM) needs to be placed on a PM which is already hosting some VMs. This paper emphasizes on subsequent placement where we are assuming that all the PMs are already hosting certain number of VMs and our task at hand is to place a newly arrived user's request (VM) to the best possible host (PM).

This paper presents a dynamic virtual machine placement algorithm in cloud datacenters which aims at energy efficiency, load balance and improved performance by reducing the need of virtual machine migrations. Our proposed placement scheme is based on a 'divide and conquer' approach combined with vector arithmetic to select the best possible PM to host an incoming VM. The proposed mechanism is compared with the widely used bin packing's First-Fit placement technique and considerable performance improvements are registered.

2. Related Work

Most of the research in resource management in cloud computing is concentrated towards virtual machine migrations. It is seen as the 'make or break' point of a datacenter's performance [1]. However, in order to improve the migration scenario, researchers need to go back and analyze when and how virtual machines are allocated to

physical machines. To start with, we studied the mechanisms for VM allocation which are discussed in [2] by comparing fixed-price and auction-based scenarios. Every modern datacenter employs a resource management policy to ensure high Quality of Services to its customers. This is possible by maintaining a healthy power-performance trade-off, where idle servers are switched-off to minimize energy consumption [3]. Most of the traditional VM placement algorithms consider placement as an instance of bin-packing problem [4, 5, and 6] and offer server consolidation as a possible solution [7, 8 and 6]. Using similar approach, ant colony optimization based consolidation algorithm is proposed and compared with greedy based algorithm in [4] and [9]. Every cloud provider strives to minimize the operational cost of a cloud system so as to maximize profits, while at the same time ensuring that the service level agreement parameters are met. The steps taken by cloud service providers and their effects on system's performance are detailed out in [1]. VM migrations, its causes and effects on datacenters can give a good starting point to further delve into VM placements. Overheads of virtual machine migrations, especially in an over-committed datacenters, are studied in [10] and an algorithm is presented to minimize migrations. Among all the different types of resources available in a cloud system, communication resources are the most crucial ones. In order to conserve them, data replication is proposed in [11] and [12] where QoS is also improved by reducing communication delays. A detailed study and comparison of 18 existing VM placement algorithms is done in [13]. There are different objectives of initial VM placement algorithms; one of them is minimizing job completion time of the incoming VM which is considered in [14] for which a knapsack-based VM placement approach is suggested. Live VM migration on Xen and KVM a comparative analysis is found and system is known PMigrate as well as the implementation is also found in [15]. Resource needs during migration and its effect on datacenters is evaluated in [16] for different set of workloads. This gives us a better understanding of the impact of VM migration on effective resource management techniques employed in a cloud system. Another approach to control unnecessary migrations is suggested in [17] which require prioritizing VMs with steady capacity and in [7] where dynamic consolidation and migration algorithm is introduced to avoid SLA violations.

A detailed research on cloud migration is carried out in [18] whereas in [19] efforts are being taken to reduce communication delays by routing the user's traffic which is close towards the resources in the datacenter. Energy-Response time Product (ERP) is used to evaluate server farm management policies for both stationary and dynamic demands in [8] whereas for predictable and time-confined loads, an optimal VM placement strategy is devised in [20]. To further reduce communication overheads between VM and its image, [21] places them on the same node thereby improving cloud's performance. Data-intensive applications are also dealt with in the same manner by keeping all data in the nearby nodes, as proposed in [22]. Based on the resource needs of virtual machines, several VMs are consolidated and provisioned together and statistical multiplexing is given in [23]. A two-level runtime reconfiguration strategy in a vector arithmetic model is used to maximize resource utilization and to reduce total migration time, includes local adjustments and parallel VM migrations [24]. On the other hand, authors in [25] devised a multi-level generalized assignment problem in order to ensure provider's profit and to constraint power budget of a virtualized datacenter. First-fit heuristic is used to provide solution to the assignment problem. On the contrary, the approach used in [5] is to consider VM placement as a classical bin-packing problem and greedy approach is used to solve it using the best-fit heuristic. With the aim of proper resource utilization and reduced energy consumption, authors in [6] have proposed a VM placement technique using a two-stage heuristic algorithm, addressing VM consolidation which may degrade the datacenter's performance, if done aggressively. After an extensive review of research in the field of VM migrations to improve datacenter's performance, there is a shift in interest towards placement which can be considered as a core area as far as performances and migrations are considered. A successful attempt in migration can improve the performance of a datacenter but it also incurs some cost in terms of efficiency and time. A multi-objective VM placement strategy is presented in [26] with the aims of minimizing

energy consumption, minimizing task waiting time while maximizing effectiveness of physical machines. In [27], authors proposed a hybrid queuing method for VM placement that ensures greater QoS to users and maximizes revenues for the service provider. Energy conservation has drawn considerable attention in cloud datacenters in the recent times. Many energy efficient VM placement techniques are proposed in [28, 29 and 30] where the main aim is reduction in power consumption by consolidating servers [29] and/or by using genetic algorithms [30, 31]. Scheduling techniques commonly used in a cloud environment are discussed in [32] with the improvisation in certain key performance factors such as response time, resource utilization, etc. Further, [33] assess cloud services based on a comparative framework which outlines the merits and demerits of multiple cloud service providers. A common management services framework for IaaS Cloud is proposed in [34] featuring cost reduction and resources usage optimization features to improve the performance metrics of cloud computing. On the other hand, [35] gives a comparative analysis tool that helps cloud users in selection of a preferable cloud infrastructure. Machine learning techniques are applied in [36] along with particle swarm optimization to classify users requests and improve the QoS experienced by the users by efficiently mapping service requests with the available resources. Many references mentioned above fail to go to the core issue behind migration which is an effective VM placement. VM placement is discussed in some references [1, 4, 5, 6, 8, 9, 12, 13, 14, 20, 21 and 25]; wherein, most of the authors keep only one or two objectives in mind such as energy efficiency or QoS maintenance [8, 12]. Most of the work done in this line considers placement as a classic bin-packing problem and tries to solve it using best-fit or first-fit approach [4, 9 and 14]. There is a need to consider placement as a multi-dimensional problem highlighting heterogeneity in physical as well as virtual machines. If we can take this heterogeneity to our advantage by making VM placement based on it, then a lot of unnecessary migrations can be saved which will further save energy and maximize resources' utilization. Our present paper is one such attempt which proposes a dynamic VM placement technique by using 'divide and conquer strategy combined with vector arithmetic. It ensures performance of cloud system, profits to cloud providers, service availability to cloud customers and also an optimum management of cloud's resources along with energy efficiency. Further, a multi-dimensional approach is considered to guarantee benefits to both the parties involved, i.e. customer and cloud service provider.

The present paper is structured in the following manner: Section 1 presents the Introduction. Section 2 represents the related work and section 3 presents the general scenario of VM placement in all virtualized datacenters, section 4 defines the VM placement problem along-with identifying the objectives of our approach. Section 5 describes the system model used, while, section 6 explains the proposed VM placement technique with algorithms and formulae used. Finally, section 7 discusses the results obtained after simulating the proposed technique and comparing it with First-Fit bin packing technique. Conclusion and future work is given in section 8 followed by references.

3. General Scenario

The VM placement scenario considered in our paper consists of cloud users, their resource requests in the form of virtual machines, a virtual machine dispatcher and a number of physical machines arranged in dynamic clusters with one machine designated as cluster head in each cluster. Clusters are dynamic in the sense that its member PMs can leave the cluster if their membership criteria changes.

Cloud datacenter consists of m physical machines or PMs which are heterogeneous in the type and amount of resources they carry. Resources considered in this paper are CPU (denoted as P), memory space (M) and I/O units (denoted as I/O). Every cloud user needs to use these resources in the amount that his application requires. Being a part of a utility computing system, a cloud computing user is also required to pay for the amount of resources used by him. In the present work, this price is considered as 'profit' for the cloud service provider. In order to gain access to a cloud's resource, a user makes a resource request to the datacenter. In return, datacenter places the user's request in one of the available physical machine provided that the resource requirements of the user are met by the resource availability of that physical machine. It may seem that this

arrangement means total number of users serviced by a datacenter is bounded by total number of physical machines in the datacenter. However, in order to cater to as many users as possible, datacenters rely on the technique of virtualization. Virtualization enables a single physical machine to be shared by multiple users by slicing its resources as per the users' requirements. This means a single machine will not cater to a single user if his resource requirements are less than the resource availability of the machine. Instead another user(s) can use the remaining resources without any conflict of interest. These VMs from different users will share a single physical machine as long as this machine has enough resources to service them. VM placement is the task of figuring out which VM will be placed at which physical machine so that the physical machine's resources are utilized in an optimum manner. An efficient VM placement maintains a uniform load among PMs so that the frequency of migrations is reduced. In the context of this paper, load refers to number of VMs hosted on a PM which is also an indication of amount of PM's resources consumed.

4. Problem Definition

Given a cloud datacenter with **m** heterogeneous physical machines and **n** heterogeneous virtual machines, VM placement refers to selecting a suitable PM to host an incoming VM such that the resource requirements of the VM are completely satisfied by the chosen PM. We understand that more than one PM may meet the resources availability criteria. In such situations, our placement technique selects the most suitable candidate PM using multi attribute utility theory. Resource requirement of each VM is represented by a demand vector (D) as shown in figure 1. As mentioned earlier, P refers to CPU demand, M refers to memory demand, I/O refers to input-output demand and RP refers to rental price which will be paid by the cloud user once his resource demands are satisfied. On the other hand, resources in a physical machine are also represented by 2 vectors, namely, resource availability vector (A) and resource utilization vector (U). These two vectors are represented in figure 2(a) and (b). In figure 2(a), PA, MA and I/OA refer to CPU availability, memory availability and I/O availability respectively. In utilization vector U shown in figure 2(b), PU, MU and I/OU refer to CPU utilized, memory utilized and I/O utilized respectively.

Objective- Our objectives, behind proposing a new VM placement technique, are-

i. **Efficient Resource Provisioning**
ii. **Dynamic Load Balancing**
iii. **Profit guarantee to the cloud provider**
iv. **Reduced VM Migrations**
v. **Effective Resource Utilization**
vi. **Faster Placement of VMs or reduced placement time**

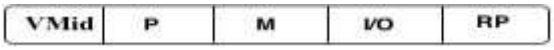

Fig 1: Demand Vector, D, of an incoming VM

Fig 2(a) Resource Availability Vector, A, **2(b)** Resource Utilization Vector, U

5. System Model

In order to implement the proposed profitability-aware VM placement, we follow the 'divide and conquer' approach by considering a datacenter consisting of six queues (Q1 to Q6) and six clusters (C1 to C6). The number of queues and/or clusters depends on the type of resources considered for placement. In general, for k types of resources there will be 2^k queues and clusters. At present we are considering three types of resources, namely CPU (MIPS), memory (MB) and I/O (B/sec), hence, 6 queues and clusters. These queues will hold incoming VMs as per the membership criteria while clusters contain PMs. Each VM's demand vector D will enter any one of the six queues based on its resource requirements. Similarly, each physical machine in the datacenter will be allotted to one of the six clusters based on its resources availability. If the resource availability of a particular PM changes, due to a VM placement or migration, the said PM will be re-allocated to a new cluster as per the resource availability/membership criteria. We consider queue Q1 to have all demand vectors (D), with highest processor demand and lowest I/O demand. Q2 consists of all VMs whose processor demand is the highest and memory demand is the lowest. Q3 consists of all VMs whose memory demand is the highest whereas processor demand is the lowest. Q4 consists of all VMs whose memory demand is

the highest whereas I/O demand is the lowest. Q5 consists of all VMs whose I/O demand is the highest whereas processor demand is the lowest. Q6 consists of all VMs whose I/O demand is the highest whereas memory demand is the lowest. Table 1 describes the membership criteria for queues and clusters. The comparison of highest or lowest resource demand of a VM is done within the vector by comparing its direction cosines and is irrespective of other incoming VMs. For example, consider a demand vector D as shown below-

VM5	20	9	25	100

Here VM_{id} is VM5, processor demand (P) is 20, memory demand (M) is 9, I/O demand (I/O) is 25 and rental price (RP) to be paid is 100. Direction cosines of vector D are as follows-

$$\cos\alpha = \frac{D_x}{D} = \frac{20}{33.25} = 0.601$$

$$\cos\beta = \frac{D_y}{D} = \frac{9}{33.25} = 0.270$$

$$\cos\gamma = \frac{D_z}{D} = \frac{25}{33.25} = 0.751$$

As is evident after calculating the direction cosines that vector D has highest I/O demand (0.751) and lowest memory demand (0.270), hence it will be allocated to queue Q6 (Refer table 1). On similar lines, all physical machines whose processor availability is the highest and I/O availability is the lowest will be clubbed together in C5. Cluster C3 contains all PMs whose processor availability is the highest and memory availability is the least. Cluster C2 contains all PMs whose memory availability is the highest and processor availability is the least. Cluster C6 contains all PMs whose memory availability is the highest and I/O availability is the least. Cluster C1 contains all PMs whose I/O availability is the highest and processor availability is the least. Finally, Cluster C4 contains all PMs whose I/O availability is the highest and memory availability is the least. As an example, consider the availability vector A as

PM7	20	50	60	4

This availability vector A belongs to PM7. Its available processing capacity is 20, memory available is 50 and I/O availability is 60. It is hosting 4 VMs already. Based on this information, PM7 will be allocated to cluster C1 as PM7 has highest I/O availability and lowest processor availability (Refer table I). As is evident, every VM's demand D in Q1 will be forwarded to cluster C5 for efficient placement. Similarly, VMs from Q2 will go to C3, VMs from Q3 will go to C2, from Q4 VMs are forwarded to C6, from Q5 VMs are forwarded to C1 and Q6 VMs are forwarded to C4 for placement. Forwarding a VM's demands to a designated cluster, based on resources availability, ensures efficient resource provisioning (objective (i)) and effective resource utilization (objective (v)). An important step to note here is that VMs, in each queue, are sorted as per the rental price (RP) offered, before forwarding them to their respective clusters. This step ensures that in each queue, VM which is offering the highest profit (or rental price) will be serviced first for placement. Hence, we term this placement technique as 'profit-aware' VM placement and this step ensures fulfillment of objective (iii). A detailed representation of our proposed system model is given in figure 3.

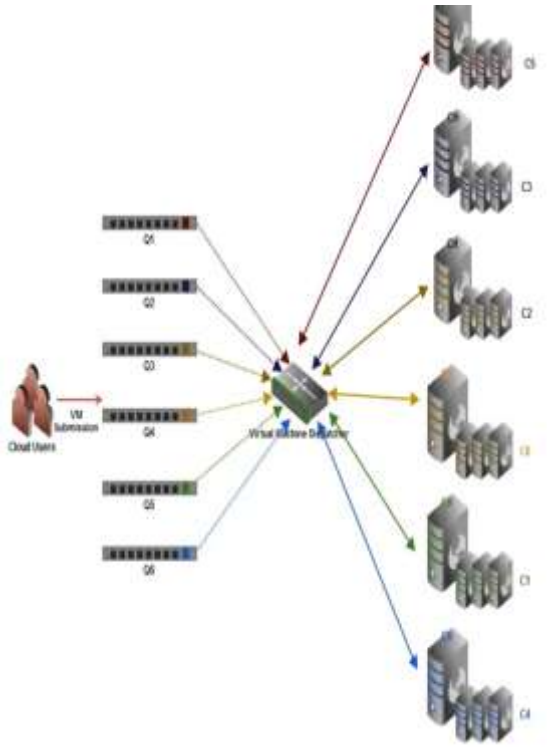

Fig 3: System Model of proposed VM

Table I: Description of Queues and clusters

Queue Name	VM's property	Linked Cluster Name	Cluster's property
Q1	Highest demand-p Lowest demand-I/O	C5	Highest availability-p Lowest availability-I/O
Q2	Highest demand-p Lowest demand-M	C3	Highest availability-p Lowest availability-M
Q3	Highest demand-M Lowest demand-p	C2	Highest availability-M Lowest availability-p
Q4	Highest demand-M Lowest demand-I/O	C6	Highest availability-M Lowest availability-I/O
Q5	Highest demand-I/O Lowest demand-p	C1	Highest availability-I/O Lowest availability-p
Q6	Highest demand-I/O Lowest demand-M	C4	Highest availability-I/O Lowest availability-M

Table II(a) and II(b) describes the various parameters which are used in proposed system model, in terms of the average resources available in each cluster and average resources held in each queue respectively. VMs are dynamically allocated to queues based on the criteria explained above.

Table II(a): Parameters of System Model used

Clusters	Average amount of resources available		
	P	M	I/O
C1	10	15	18
C2	7	11	8
C3	18	12	14
C4	17	17	18
C5	13	11	8
C6	10	14	9

Table II(b): Parameters of System Model used

Queue	Average amount of resources held		
	P	M	I/O
Q1	8	5	2
Q2	2	2	2
Q3	2	8	4
Q4	4	9	1
Q5	6	7	8
Q6	2	1	9

6. Proposed VM Placement Technique

Profitability-aware VM placement starts with cloud users who submit their VMs' demands (D) in one of the designated queues dynamically. Every queue is then sorted in decreasing order of the rental price offered in the demand vector. An entity called virtual machine dispatcher (VMD) fetches the front VM's D vector from each queue and forwards it to its designated cluster. Every cluster has a nominated cluster-head (CH) which receives the incoming VM's D vector and compares it with the resource availability vectors (A) of all the physical machines in the cluster. These availability vectors are updated by PMs and submitted to the cluster-head periodically. All the PMs, whose availability vectors (A) satisfy the incoming VM's demand vector (D), are inserted in a candidate queue (CQ). CQ represents feasible PMs who can provide the service to the incoming VM comfortably. Now, the placement task is reduced to finding out the best possible PM among the chosen candidate PMs who can create the VM without unbalancing the cluster's load. For this, the average load density (ALD) of a cluster is calculated after considering that the incoming VM is placed in one of the candidate PMs. Likewise the machine load density (MLD) of every candidate PM is also calculated after assuming that the incoming VM is placed on that PM. The method for calculating ALD and MLD is specified below.

$$ALD(of\ a\ cluster) = \frac{Total\ VMs\ Placed}{Total\ number\ of\ PMs\ in\ the\ cluster}$$

$$MLD(PM_i) = \frac{resources\ consumed\ on\ PM_i\ after\ VM\ placement}{resources\ available\ on PM_i\ after VM\ placement}$$

A candidate PM, whose calculated MLD is closest to its cluster's ALD after placement, is chosen as the host for the incoming VM. Comparing MLD of candidate PMs with the cluster's ALD ensures that placing VM on a host does not imbalance the overall load of the cluster. This step satisfies objective (ii) as mentioned above in section 4. Given below are the algorithms used in our proposed placement scheme.

Algorithm 1: VM's demand vector (D) submission mechanism in a queue

```
1.  for each (incoming VM's demand vector D)
      { do
        {if(max(P,M,I/O)==P&&min(P,M,I/O)==
        I/O)
          enqueue(Q1, D )
        else
if(max(P,M,I/O)==P&&min(P,M,I/O)== M)
                enqueue(Q2, D )
```

else
if(max(P,M,I/O)==M&&min(P,M,I/O)== P)
enqueue(Q3, D)
else
if(max(P,M,I/O)==M&&min(P,M,I/O)== I/O)
enqueue(Q4, D)
else
if(max(P,M,I/O)==I/O&&min(P,M,I/O)== P)
enqueue(Q5, D)
else enqueue(Q6, D)
} }

2. for each queue, sort in descending order of rental price, RP, offered in D.

Algorithm 2: VM forwarding mechanism performed by VMD. VMD retrieves the front VM's D from each queue and forwards it to its designated cluster as

1. if(dequeue(Q1,D))
forward D to C5
else if(dequeue(Q2,D))
forward D to C3
else if(dequeue(Q3,D))
forward D to C2
else if(dequeue(Q4,D))
forward D to C6
else if(dequeue(Q5,D))
forward D to C1
else forward D to C4

Algorithm 3: Selection of Candidate PMs in each cluster and subsequent VM placement, performed by CH

1. Calculate ALD (of each cluster) = Total Resources Utilized/Total Resources Available
2. foreach(incoming VM's demand D)
do {foreach(PM's availability vector A)
do {if(P<=PA&&M<=MA&&I/O<=I/OA)
enqueue (CQ, PM)
else discard PM } }
3. if(CQ has more than 1 PM)
{foreach(PM in CQ)
do {/*Calculate MLD of PM after assuming VM placement*/
MLD(PM)=resources consumed/
resources available }
if(|ALD-MLD(PM_i)|<|ALD-MLD(PM_j)|)
VM is placed on PM_i
else VM is placed on PM_j
/*i and j are subscripts for candidate PMs in CQ*/ }
elseif(CQ has only one PM)
place incoming VM on this PM
else discard VM /*indicating no suitable PM for placement in the cluster*/

Figure 4 below shows the time-line diagram of the above explained placement procedure.

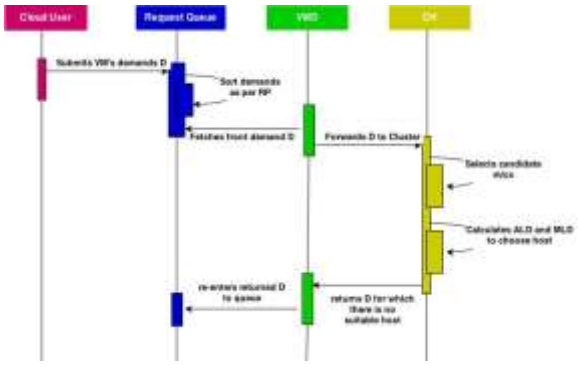

Fig 4: Timeline of the proposed VM placement technique

7. Results and discussion

The simulation scenario of proposed VM placement technique consists of a cloud datacenter with 12 physical machines and 10 virtual machines. Resource demands of VMs for P, M and I/O range from 0 to 20 instances. Similarly, resources availabilities of PMs are ranging from 0 to 20 instances. The initial resources utilization of each PM is given and further after each VM placement it is calculated by subtracting D from A. A snapshot of the example considered for simulation is given below-

	Resource Demands				
VM IDs	P	M	I/O	RP	Queue Alloted
VM1	7	5	3	20	Q1
VM2	2	1	9	30	Q6
VM3	4	5	6	10	Q5
VM4	4	9	1	80	Q4
VM5	9	10	10	50	Q5
VM6	1	7	5	40	Q3
VM7	6	7	8	70	Q5
VM8	1	9	3	10	Q3
VM9	2	2	2	100	Q2
VM10	8	5	1	90	Q1

Cluster Alloted	PM IDs
C2	PM1
C1	PM2
C3	PM3
C5	PM4
C5	PM5
C2	PM6
C2	PM7
C3	PM8
C4	PM9
C5	PM10
C1	PM11
C6	PM12

Fig 5: Snapshot of VMs, PMs and their allocated clusters

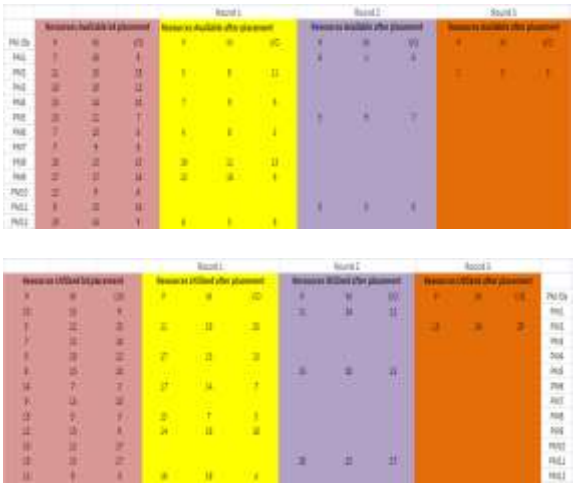

Fig 6: Snapshots of resources availability and utilization

Figure 7 gives the individual number of VMs placed in each round of proposed VM placement technique. All the VMs are successfully placed in three rounds, with the placement distribution as 6, 3 and 1 in rounds 1, 2 and 3 respectively. Figures 8, 9 and 10 give resource utilization in each cluster for CPU, memory and I/O respectively before and after VM placements. As is evident from the figures, all the three resource types register a better utilization after applying the proposed placement.

The consumption of resources for each cluster cluster is shown in Figure 11. It also shows the total resource utilization increments after a successful Virtual Machine placement. Figure 12 shows the total utilization of resources of each physical machine in the datacenter. One of the primary objectives behind devising this placement technique was to consume or use the datacenter's resources in the best possible manner. As is evident from the figure 12 below, our claim of optimum resource utilization is successfully proved.

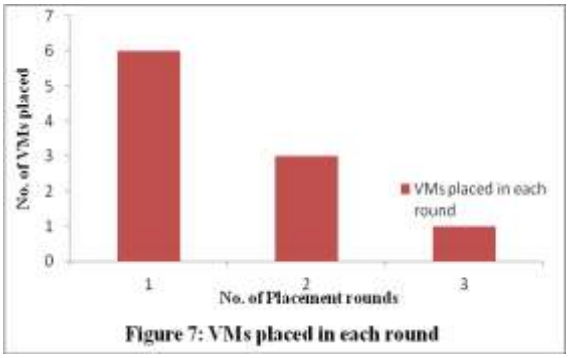

Figure 7: VMs placed in each round

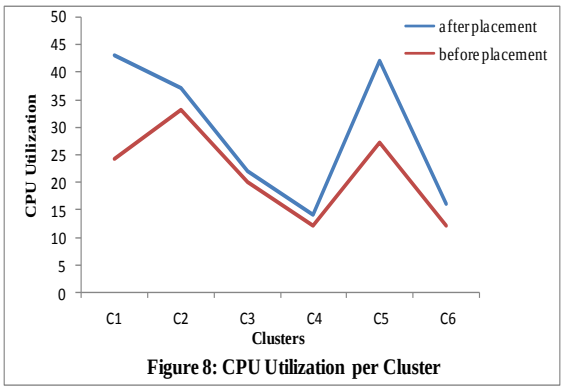

Figure 8: CPU Utilization per Cluster

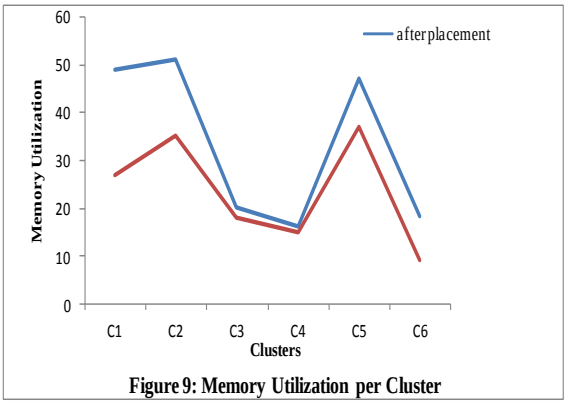

Figure 9: Memory Utilization per Cluster

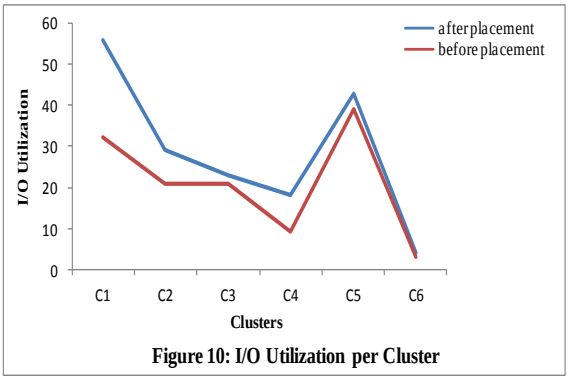

Figure 10: I/O Utilization per Cluster

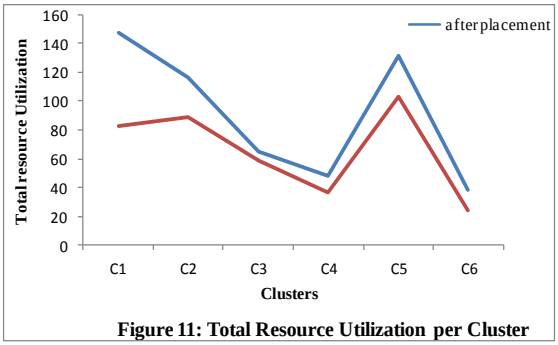

Figure 11: Total Resource Utilization per Cluster

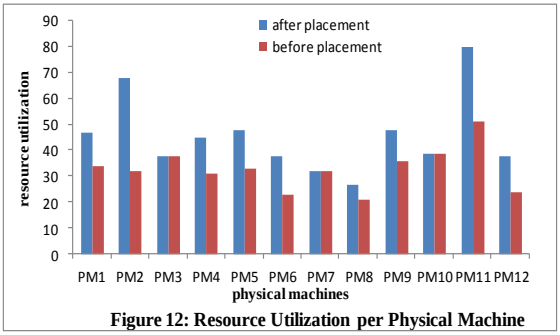

Figure 12: Resource Utilization per Physical Machine

Figure 13 and 14 show the average load density per cluster and on each physical machine respectively. Our objective is to maintain a uniform load on physical machines in each cluster. From figure 14, we see that cluster 1 consisting of two machines, PM2 and PM11, have similar loads. Physical machines PM1, PM6 and PM7 belong to cluster 2 and they carry close loads if not same. All the machines in cluster 3, PM3 and PM8, share a similar load structure. Similarly, PM4 and PM10 in cluster 5 carry an even load if not equal. Please note that our aim was not to give equal load (in figurative sense) to each physical machine inside a cluster, instead we are trying to maintain a closer load density on each machine in a cluster, i.e., avoiding stark differences in the number of VMs or amount of resources consumed within a cluster. Figure 15 depicts the revenues earned by each cluster. It is also considered as profit for the cloud provider and is calculated in terms of the rental price offered by the user when his VM gets placed in a cluster. By categorizing PMs into clusters and VMs requests into queues, we have streamlined the entire placement procedure and followed the 'divide and conquer' strategy. This reduces the time required to place a VM on a PM, when compared to 'first come first serve' strategy of VM placement, as shown in figure 16. Figure 17 compares the total resource utilization between First-Fit and our proposed technique which is better than First-Fit in respect of placement of virtual machines.

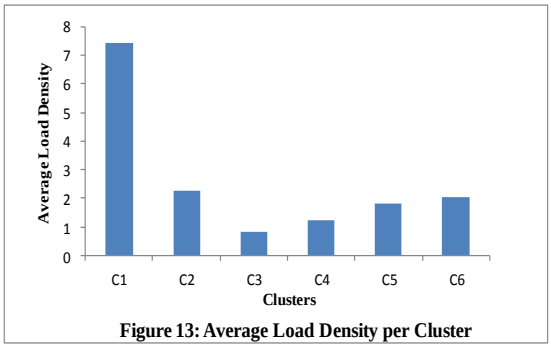

Figure 13: Average Load Density per Cluster

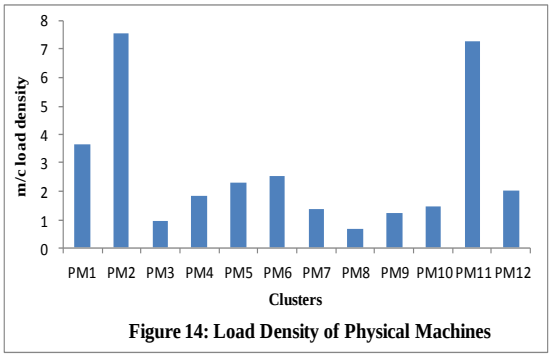

Figure 14: Load Density of Physical Machines

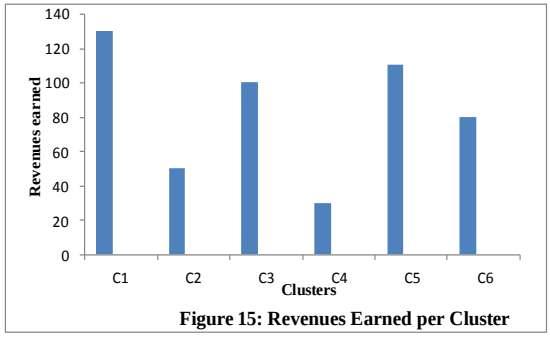

Figure 15: Revenues Earned per Cluster

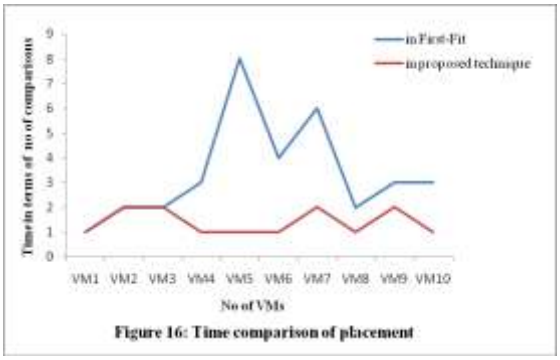

Figure 16: Time comparison of placement

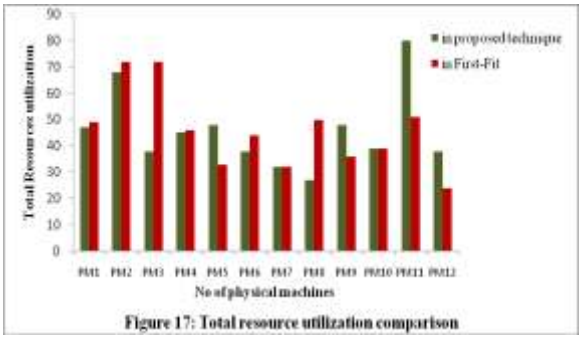

Figure 17: Total resource utilization comparison

8. Conclusion and Future Scope

VM place is an important issue in cloud datacenters. An efficient placement streamlines the cloud users' tasks in the best possible manner which enhances the performance and serviceability of cloud computing. Related work on VM placement mostly revolves around in packing technique and considers optimal resource utilization as their main objective. In the present research paper, in addition to efficient resource utilization, the proposed VM placement is also a 'profit-aware' technique which ensures that the VM offering the highest rental price in each queue gets the priority over others for placement. By using divide and conquer approach, our placement technique chooses a suitable host PM in such a manner that the cluster's load is not disturbed. This, in turn, helps in reducing VM migrations and saves energy. Our proposed algorithm is compared with first-fit variant of bin packing and the obtained experimental results looks promising.

Future scope includes comparing the proposed work with other successful placement techniques. Also, dynamic clustering methods need to be looked upon to improve the performance of the proposed placement approach.

References

1. Yue Gao, Yanzhi Wang, Gupta, S.K., Pedram, M., An energy and deadline aware resource provisioning, scheduling and optimization framework for cloud systems, Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2013 International Conference on, Sept. 29 2013-Oct, p. 1-10.
2. Zaman, S., Grosu, D., Combinatorial Auction-Based Allocation of Virtual Machine Instances in Clouds, Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, Nov. 30 2010-Dec. 3 2010, p.127-134.
3. Beloglazov, A, Buyya, R., Energy Efficient Allocation of Virtual Machines in Cloud Data Centers, Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on , 17-20 May 2010, p. 577-578.
4. Feller, E., Rilling, L., Morin, C., Energy-Aware Ant Colony Based Workload Placement in Clouds, Grid Computing (GRID), 2011 12th IEEE/ACM International Conference on , 21-23 Sept. 2011, p. 26-33.
5. Jiankang Dong, Xing Jin, Hongbo Wang, Yangyang Li, Peng Zhang, Shiduan Cheng, Energy-Saving Virtual Machine Placement in Cloud Data Centers, Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on , 13-16 May 2013, p. 618-624.
6. Jiankang Dong, Hongbo Wang, Xing Jin, Yangyang Li, Peng Zhang, Shiduan Cheng, Virtual Machine Placement for Improving Energy Efficiency and Network Performance in IaaS Cloud, 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW), p. 238-243.
7. Bobroff, N., Kochut, A. and Beaty, K., Dynamic Placement of Virtual Machines for Managing SLA Violations., in 'Integrated Network Management' , 2007 IEEE, p. 119-128.
8. Anshul Gandhi, Varun Gupta, Mor Harchol-Balter, Michael A. Kozuch, Optimality analysis of energy-performance trade-off for server farm management, Performance Evaluation, November 2010, Volume 67, Issue 11, Performance 2010, p. 1155-1171.

9. Yongqiang Gao, Haibing Guan, Zhengwei Qi, Yang Hou, Liang Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, Journal of Computer and System Sciences, Volume 79, Issue 8, December 2013, p. 1230-1242, ISSN 0022-0000.
10. Xiangliang Zhang, Zon-Yin Shae, Shuai Zheng, Jamjoom, H., Virtual machine migration in an over-committed cloud, Network Operations and Management Symposium (NOMS), 16-20 April 2012 IEEE, p. 196-203.
11. Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., Zomaya, A Y., Energy-efficient data replication in cloud computing datacenters, Globecom Workshops (GC Wkshps), 9-13 Dec. 2013 IEEE, p. 446-451.
12. Jenn-Wei Lin, Chien-Hung Chen, Chang, J.M., QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems, Cloud Computing, IEEE Transactions on , Jan.-June 2013, vol.1, no.1, p. 101-115.
13. Mills, K., Filliben, J., Dabrowski, C., Comparing VM-Placement Algorithms for On-Demand Clouds, Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on , Nov. 29 2011-Dec. 1 2011, p. 91-98.
14. Kangkang Li, Huanyang Zheng, Jie Wu, Migration-based virtual machine placement in cloud systems, Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on , 11-13 Nov. 2013, p. 83-90.
15. Xiang Song, Jicheng Shi, Ran Liu, Jian Yang, and Haibo Chen, Parallelizing live migration of virtual machines, in Proceedings of the 9th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments (VEE '13), ACM, New York, NY, USA, p. 85-96.
16. Senthil Nathan, Purushottam Kulkarni, and Umesh Bellur, Resource availability based performance benchmarking of virtual machine migrations, in Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE '13), Seetharami Seelam (Ed.), ACM, New York, NY, USA, p. 387-398.
17. Tiago C. Ferreto, Marco A. S. Netto, Rodrigo N. Calheiros, César A. F. De Rose, Server consolidation with migration control for virtualized data centers, Future Generation Computer Systems, 2011, vol. 27, issue 8, p. 1027-1034.
18. Jamshidi, P., Ahmad, A., Pahl, C., Cloud Migration Research: A Systematic Review, Cloud Computing, IEEE Transactions on, July-December 2013, vol.1, no.2, p. 142-157.
19. Joseph Doyle, Robert Shorten, Donal O'Mahony, Stratus: Load Balancing the Cloud for Carbon Emissions Control, *IEEE* Transactions on Cloud Computing, 2013, vol.1, no. 1, p. 1.
20. Wubin Li, Johan Tordsson, and Erik Elmroth, Virtual machine placement for predictable and time-constrained peak loads, in Proceedings of the 8th international conference on Economics of Grids, Clouds, Systems, and Services (GECON'11), Vanmechelen, Jörn Altmann, and Omer F. Rana (Eds.), Springer-Verlag, Berlin, Heidelberg, p. 120-134.
21. Xiaoqiao Meng, Canturk Isci, Jeffrey Kephart, Li Zhang, Eric Bouillet, and Dimitrios Pendarakis, Efficient resource provisioning in compute clouds via VM multiplexing, in Proceedings of the 7th international conference on Autonomic computing (ICAC '10), ACM, New York, NY, USA, p. 11-20.
22. Wei Chen, Xiaoqiang Qiao, Jun Wei, Tao Huang, A Profit-Aware Virtual Machine Deployment Optimization Framework for Cloud Platform Providers, Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, 24-29 June 2012, p. 17-24.
23. Weiming Shi, Bo Hong, Towards Profitable Virtual Machine Placement in the Data Center, Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on, 5-8 Dec. 2011, p. 138-145.
24. Hua-Jun Hong, De-Yu Chen, Chun-Ying Huang, Kuan-Ta Chen, Cheng-Hsin Hsu, QoE-aware virtual machine placement for cloud games, Network and Systems Support for Games (NetGames), 2013 12th Annual Workshop on , 9-10 Dec. 2013, p. 1-2.
25. Mishra, M., Sahoo, A., On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach, Cloud Computing (CLOUD), 2011 IEEE International Conference on, 4-9 July 2011, p. 275-282.

26. H. Y. Kao, Y. M. Yang and C. H. Huang, Dynamic virtual machines placement in a cloud environment by multi-objective programming approaches, 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, 2015, p. 364-365.
27. S. K. Addya, A. K. Turuk, B. Sahoo and M. Sarkar, A hybrid queuing model for Virtual Machine placement in cloud data center, 2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), Kolkata, 2015, p. 1-3.
28. P. Wattanasomboon and Y. Somchit, Virtual machine placement method for energy saving in cloud computing, 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, 2015, p. 275-280.
29. N. Khalilzad, H. R. Faragardi and T. Nolte, Towards Energy-Aware Placement of Real-Time Virtual Machines in a Cloud Data Center, High Performance Computing and Communications (HPCC), New York, NY, 2015, p. 1657-1662.
30. D. Liu, X. Sui and L. Li, An energy-efficient virtual machine placement algorithm in cloud data center, 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, 2016, p. 719-723.
31. S. Telenyk, E. Zharikov and O. Rolik, An approach to virtual machine placement in cloud data centers, 2016 International Conference Radio Electronics & Info Communications (UkrMiCo), Kiev, Ukraine, 2016, p. 1-6.
32. Sanaz Yousefian and Ahmad Habibi Zadnavin: Scheduling Virtual Machines in Cloud Computing. MAGNT Research Report (ISSN. 1444-8939) Vol.3 (1). P. 389- 397.
33. Farrukh Nadeem: Towards Comparative Evaluation of Cloud Services. MAGNT Research Report (ISSN. 1444-8939) Vol.2 (5):PP.61-68. DOI: dx.doi.org/14.9831/1444-8939.2014/2- 5/MAGNT.8.
34. Narander Kumar and Shalini Agarwal, Managing IaaS Cloud Using Common Management Services Framework. MAGNT Research Report (ISSN. 1444-8939) Vol.2 (5).PP:189-198.
35. Narander Kumar and Shalini Agarwal, QoS based Enhanced Model for Ranking Cloud Service Providers. MAGNT Research Report (ISSN. 1444-8939) Vol. 2(6).PP:32-39.
36. Narander Kumar and Pooja Patel, Maintaining the QOS Using FANN Mechanism with PSO in Cloud Computing Environment. MAGNT Research Report (ISSN. 1444-8939) Vol. 4(2).PP:87-99.