IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 3, March 2016

A Load Balancing Approach to Minimize the Resource Wastage in Cloud Computing

Sachin Soni[1], Praveen Yadav[2]

Department of Computer Science, Oriental Institute of Science and Technology, Bhopal, India[1,2]

Abstract: Cloud computing is emerging as a new paradigm for the IT industries and it become so popular in a very short time. Virtualization is one of the core technologies in the cloud and enables the provider to deploy infrastructure services in cloud environment. Cloud provides the computing resources to the client as a service in the form of VM. VM is the logical entity which is similar to the PM and executing the user application. As the demands for the computing resources is increasing, proper resource consumption of the physical resources main challenge for the service provider. Since cloud resources are communal by the numerous users and the demands for the resources is change frequently, so there is a requirement for a valuable load balancing method that enlarge the resource utilization and amplify cloud services the performance. In the past few decades, various load balancing method have been proposed. These all methods use the VM migration approach for balance the load on the PM. Since cloud consist of several type resources i.e., CPU, memory and bandwidth, so if one type resource is utilize more than other resource will be wastage. This resource wastes diminish the resource utilization. This paper proposed a load balancing approach that amplify the physical resource utilization and curtail the energy consumption. To calculate the performance of the proposed approach it is compare with the existing load balancing approach and judge against the number of migration, energy consumption. CloudSim simulator is use as a simulation tool to create the cloud environment. Experiment results say that proposed approach gives better result as compare to the existing load balancing method.

Keywords: Cloud, virtual machine, physical machine, energy efficient, resource utilization, VM migration.

I. INTRODUCTION

Cloud computing is one of the fastest growing technology in the industry as well as society [1, 2]. It is not a completely new concept; it is originated for the grid computing and distributed computing [3]. One of the main advantages of the cloud service is that use these services anywhere at any time [4]. Moreover cloud is a utility model where user pays only for the used resources. It provides on demand resources to the user as a service. Cloud support three types of services named software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and these services can be deploy in three different way i.e., private, public and hybrid.

Virtualization [6, 7] play a vital role in cloud computing. It is an interface which separate hardware from software and provides the benefits of server consolidation and live migration. A main benefit of the virtualization is that, it allows creating multiple virtual machines (VMs) on a single physical machine (PM) and migrating them to the other PM when the PM is overloaded or under loaded [2, 3]. Hypervisor is the program which is use for the virtualization. It creates the VM based on the user needs and hand over to the user for executing their application. VM is the logical entity that uses the PM resources for executing the user application and user think that VM is running in their own PM. Since, in cloud number of VM can run in a single PM so single resource is shared by the multiple users. Hence use of the virtualization increases the resource utilization but introduce new challenge i.e., proper load balancing. Performance of the PM is depends on the load. It the PM is overloaded then the performance of all the VM running on that PM is degrades. Due to this reason load balancing is the core function of the cloud management. Load balancing in the cloud is a critical task due to the dynamic change in VM requirement [5].

A load balancing approach can be static or dynamic. The static approach is one which where the system information is not essential and working is less complex comes under static scheme and the one which brings additional cost to the system but having the feature by which state of the system can be change comes under the dynamic scheme. Static approach is more suitable for the cloud because of its dynamic behavior. In the static load balancing approach fixed lower and upper thresholds are use and the load of the PM between the lower and upper threshold. The values of lower and upper threshold define the overutilized or overloaded and underutilize or underloaded situations. If load on the PM is below the lower threshold then server is under utilize whereas load greater than the upper threshold says that PM is overloaded.

To deal with the overloaded and under loaded situation VM migration is used which migrate the VM form one PM to another PM. VM migration [3] is the main feature of the virtualization and its enable the provider to deal with the server consolidation, load balancing, server failure and hot spot mitigations.

During the survey of VM migration theory it is found that number of migration affect the system performance. System performance decreases with increasing in number of migrations. These numbers of migration can be minimized by the proper VM scheduling approach. Moreover number of running server or active server is also

DOI 10.17148/IARJSET.2016.3322

depends on the effective VM scheduling approach. Previous study [2, 3] says that energy consumption is depends on the number of running server. Due to this reason a load balancing approach must minimize the number of migrations and running servers.

The load balancing approach is depends on the thresholds. Lower and upper thresholds are use to represent the underloaded and overloaded situation respectively. This paper proposed a load balancing approach which set the value of upper and lower threshold on the CPU utilizations. To analyze the performance of the proposed load balancing it is compare with the existing approach.

II. LITERATURE SURVEY

Load balancing is the core management function of the cloud because the efficiency of the cloud services is heavily depends on the load balancing approach. Due to this reason several load balancing approach have been proposed in the last few decades.

R. Addawiyah et al. [8], proposed a migration based load balancing method for the cloud. This approach uses two bounds and migrate the VM according to these bounds values. The values of lower bound and upper bound are 10 and 90 respectively.

If the current resource’s CPU usage is greater than 90% (which means that it is overloaded), then the smallest VM will be migrated to the other PM whose CPU utilization is less than 50%. If the current resource’s CPU usage is less than 10% (which means that it is underutilized), then the all VM will be migrated to the other PM whose CPU utilization is less than 70%. After selecting the PM, selected VM will be place to the lower utilize PM. Limitation of this approach is that it may increase the number of migrations.

E. Gupta et al. [9] introduced an Ant Colony Optimization technique based on load balancing. This technique notice overloaded and under loaded servers and performs load balancing between identified servers of data center. By using this technique we can achieve availability, effective resource utilization, cloud handled maximum number of requests and minimize the response time.

D. Gmach [10] et al. describes about the threshold based approach a threshold based reactive approach to dynamic workload handling. The paper has tired finding the underwhelming and overwhelming situation for the PM and initiates migration as essential. This approach is not much suited for IaaS environment.

A. Beloglazov et al. [11], introduced an approach for balancing the load in the cloud. The proposed energy-aware heuristics allocation provisions cloud resources to the client for running the applications in a way that save energy efficiency of the data center, while delivering the negotiated Quality of Service (QoS). In this method are applied double threshold values for balancing the load. When the utilization is beyond the upper bound system is overwhelming system. Similarly when the utilization is beyond the lower bound system is called to be overwhelming.

III. RELATED WORK

The wastage of the cloud resources is core issue in the cloud and must be solved to maximizing the cloud resources. If the resource wastage is not knob properly then it will diminish the resource utilization. To diminish the resource utilization Mishra [13] et al. proposed a model that can manage the resource depletion of the cloud. But this approach present only theory on the VM placement and not implemented in any environment to compute the performance of the given approach. This paper presents an load balancing method to minimize the wastage of the resources. Main critical resources of the cloud are the CPU, bandwidth and memory. So in our proposed approach we considered these three resources for placing and balancing the PM.

Main goal of our approach is to minimize the resource wastage and improving the quality of the services running on the PM. For this purpose we separate the PM according to the CPU, memory and bandwidth utilization. Since we assumed three resources, so PM can be separated into six groups named CMB, CBM, MCB,MBC, BMC and BCM. Where C stands for the CPU, M stands for the memory and B stands for bandwidth. CMB type PM utilizing more CPU and then memory and then bandwidth. Similarly CBM PM usage more CPU and then bandwidth and then memory and so on. In the same way VM also into six groups named CMB, CBM, MCB, MBC, BMC and BCM. Where C stands for the CPU, M stands for the memory and B stands for bandwidth. CMB type VM utilizing more CPU and then memory and then bandwidth. Similarly CBM, VM usage more CPU and then bandwidth and then memory and so on. Proposed load balancing method s divided into two phase. In the first phase we separate the PM and VM into the six groups according to the CPU, memory and bandwidth utilizations. In the second phase we place the VM to the PM. In this phase first we determine the VM type and put it in the opposite PM group. If there are several VM in the group then VM is put to the PM where the increment in power is minimum before and after putting the VM.

Figure 1 shows the flow diagram for the proposed approach.

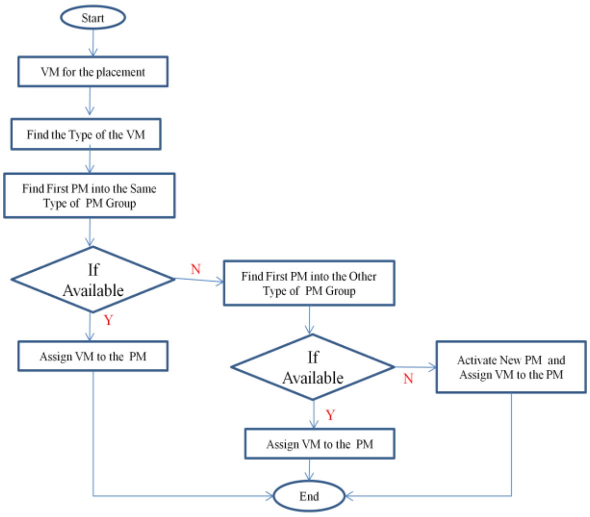
Start
VM for the placement
Find the Type of the VM
Find First PM into the Same Type of PM Group
If Available
N
Y
Find First PM into the Other Type of PM Group
Assign VM to the PM
If Available
N
Activate New PM and Assign VM to the PM
Y
Assign VM to the PM
End

Figure 1: Flow diagram for the Proposed Method

DOI 10.17148/IARJSET.2016.3322

IV. DEFINE THE TYPE OF PM AND VM

In our approach we separate the PM and VM in to 6 cluster according to the CPU, memory and bandwidth utilization. These six cluster are CMB, CBM, MCB,MBC, BMC and BCM. Where C stands for the CPU, M stands for the memory and B stands for bandwidth. CMB type PM utilizing more CPU and then memory and then bandwidth. Similarly CBM PM Usage more CPU and then bandwidth and then memory and so on. In the same way VM also into six groups named CMB, CBM, MCB, MBC, BMC and BCM. Where C stands for the CPU, M stands for the memory and B stands for bandwidth. CMB type VM utilizing more CPU and then memory and then bandwidth. Similarly CBM, VM usage more CPU and then bandwidth and then memory and so on.

Following equation is used to find the type of cluster for PM and VM.

$$VM_{CPU}^{util} = \frac{Requested \;\; CPU}{Toatl \;\; CPU \; available}$$

$$VM_{RAM}^{util} = \frac{Requested \;\; RAM}{Toatl \;\; available \;\; memory}$$

$$VM_{BW}^{util} = \frac{Requested \;\; BW}{Toatl \;\; available \;\; BW}$$

If n VM are running on the j^th PM then

$$PM\ (j)_{CPU}^{util} = \frac{\sum_{i=1}^{n} VM(i)_{CPU}^{util}}{Total \;\; CPU \;\; of \;\; the \;\; PM}$$

$$PM\ (j)_{RAM}^{util} = \frac{\sum_{i=1}^{n} VM\,(i)_{RAM}^{util}}{Total \;\; memory \;\; of \;\; the \;\; PM}$$

$$PM(j)_{BW}^{util} = \frac{\sum_{i=1}^{n} VM\,(i)_{BW}^{util}}{Total \;\; BWof \;\; the \;\; PM}$$

After calculating the utilization of CPU, memory and bandwidth next we find the PM and VM type. Following equation is used to find the type.

Types of the VMs are find as follow:
If $VM_{CPU}^{load} > VM_{RAM}^{load} > VM_{BW}^{load}$ then type of VM is CMB
If $VM_{CPU}^{load} > VM_{BW}^{load} > VM_{RAM}^{load}$ then type of VM is CBM
If $VM_{RAM}^{load} > VM_{BW}^{load} > VM_{CPU}^{load}$ then type of VM is MBC
If $VM_{RAM}^{load} > VM_{CPU}^{load} > VM_{BW}^{load}$ then type of VM is MCB
If $VM_{BW}^{load} > VM_{CPU}^{load} > VM_{RAM}^{load}$ then type of VM is BCM
If $VM_{BW}^{load} > VM_{RAM}^{load} > VM_{CPU}^{load}$ then type of VM is BMC

Types of the PMs are find as follow:
If $PM_{CPU}^{load} > PM_{RAM}^{load} > PM_{BW}^{load}$ then type of PM is CMB
If $PM_{CPU}^{load} > PM_{BW}^{load} > PM_{RAM}^{load}$ then type of PM is CBM
If $PM_{RAM}^{load} > PM_{BW}^{load} > PM_{CPU}^{load}$ then type of PM is MBC
If $PM_{RAM}^{load} > PM_{CPU}^{load} > PM_{BW}^{load}$ then type of PM is MCB
If $PM_{BW}^{load} > PM_{CPU}^{load} > PM_{RAM}^{load}$ then type of PM is BCM
If $PM_{BW}^{load} > PM_{RAM}^{load} > PM_{CPU}^{load}$ then type of PM is BMC

V. LOAD BALANCING APPROACH

Accurate load balancing method is one of the useful way to reduce the wastage of resource in cloud. A load balancing method is a three step method. These three steps are overload or underloaded PM selection, selection of VM and selection of detonation PM. In the developed load balancing method first we add the PM into the corresponding group.

a) Selecting Source PM

If the PM is over utilize or under utilize then the PM is picked as a source PM for the migration. For selecting the source PM lower and upper bound is use. If the PM utilization is below the lower bound and higher than the upper bound then PM is picked as a source PM. In this approach we set the value of lower bound and upper bound is 20 and 80.

b) Selecting VM

After deciding the source PM we transfer VM from the selecting PM. All VM is move to other PM if its utilization is below the lower bounding the source PM we transfer VM from the selecting PM. All VM is move to other PM if its utilization is below the lower bound. Similarly if its utilization is above the lower bound then we move the VM volume is greater than the utilization and upper bound difference.

c) Selecting PM for placing VM

This the most difficult task in the load balancing method. In our approach to placing the VM first we calculate the VM type then put it to the opposite PM cluster. If there are multiple PM in the selected cluster then VM is put to the PM which producing less power growth before and after placing the VM. Following algorithm is used to put VM to PM.

Algorithm 1: PM Categorization Algorithm

```
Number of Host = N
Since we consider three type of resources i.e., CPU,
RAM(Memory) and BW(Network)) ,
so the number of Host_List = 6.
[1]  First step is to create six empty Host_List.
[2]  for i=1 to N
[3]  for i=1 to 6
            if Host[i] resources have CPU≥MEM≥BW then
put it into the  CMB Host_List.
            break
            elseif
if Host[i] resources have CPU≥BW≥MEM then put it into
the  CBM Host_List.
            break
            elseif
            if Host[i] resources have MEM≥BW ≥CPU then
put it into the  MBC Host_List.
            break
            elseif
if Host[i] resources have MEM≥CPU ≥ BW then put it
into the  MCB Host_List.
            break
            elseif
if Host[i] resources have BW≥MEM≥CPU then put it into
the  BRC Host_List.
            break
            else
            if Host[i] resources have BW≥CPU≥MEM then
put it into the  BCM Host_List.
            break
[4]  End for
[5]  End for
```

DOI 10.17148/IARJSET.2016.3322

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

Algorithm 2: VM Allocation Algorithm

```
1.  Input: hostList, vmList Output: allocation of VMs
2.  for each vm in vmList do
3.  if VM= new VM then
4.  allocatedHost ← Null
5.  for each host in hostList do
6.  if hUtill< upper_threshold && host has enough resource for vm then
7.  hUtil ← h.getUtil()
8.  Assign vm to the host
9.  hUtil1 ← h.getUtil()
10. temp ←hUtil1- hUtil
11. end if
12. Assign vm to the host where host utilization difference between before and after is minimum.
13. allocatedHost ← host
14. end for
15. Update the following parameters.
    i) Chosen_Host_CPU= Chosen_Host_CPU - VM_CPU
    ii)Chosen_Host_MEM= Chosen_Host_MEM - VM_MEM
    iii)Chosen_Host_BW= Chosen_Host_BW - VM_BW
16. if Chosen_Host parameter does not belong to the current Chosen_Host_List then
    a) Remove the Chosen_host from the Current Host_List and find the new Host_List to which Chosen_Host belong as a Chosen_Host_List.
    b) Add the Chosen_Host at appropriate position in the new Chosen_Host_List
17. else
18. Adjust the Chosen_Host in current Chosen_Host_List at appropriate position
19. end if
20. if allocatedHost ← Null then
21. Active new host and assign VM.
22. else
23. vmType ← Find type of the VM (CMB, CBM, MBC, MCB, BCM and BMC)
24. Find host into opposite pmList to the vmType
25. if multiple host in the selected hostList then
26. allocatedHost ← Null
27. for each host in hostList do
28. if hUtill< upper_threshold && host has enough resource for vm then
29. hUtil ← h.getUtil()
30. Assign vm to the host
31. hUtil1 ← h.getUtil()
32. temp ←hUtil1- hUtil
33. end if
34. Assign vm to the host where host utilization difference between before and after is minimum.
35. allocatedHost ← host
36. end for
37. Update the following parameters.
    i) Chosen_Host_CPU= Chosen_Host_CPU - VM_CPU
    ii)Chosen_Host_MEM= Chosen_Host_MEM - VM_MEM
    iii)Chosen_Host_BW= Chosen_Host_BW - VM_BW
38. if Chosen_Host parameter does not belong to the current Chosen_Host_List then
    c) Remove the Chosen_host from the Current Host_List and find the new Host_List to which Chosen_Host belong as a Chosen_Host_List.
    d) Add the Chosen_Host at appropriate position in the new Chosen_Host_List
39. else
40. Adjust the Chosen_Host in current Chosen_Host_List at appropriate position
41. end if
42.     if allocatedHost ← Null then
43. Find host into remaining pmList
44. if multiple host in the selected hostList then
45. goto step 25
46. if allocatedHost ← Null then
47.     Active new host and assign VM.
48. end for
```

VI. EXPERIMENTAL RESULTS

To measure the correctness of the proposed methods it is compared with the load balancing method [8]. CloudSim simulator [12] is used for simulating the both approaches. Both approaches are compared in migration, energy consumption, and resource utilization. MIPS, RAM and bandwidth for the created VM is 2000, 10000 MB and 100000 respectively. Similarly MIPS, RAM and bandwidth for the created PM is 250, 500, 750, 1000 MIPS, 2048, 512, 128 MB of RAM and 2500, 7500, 12500 bandwidth.

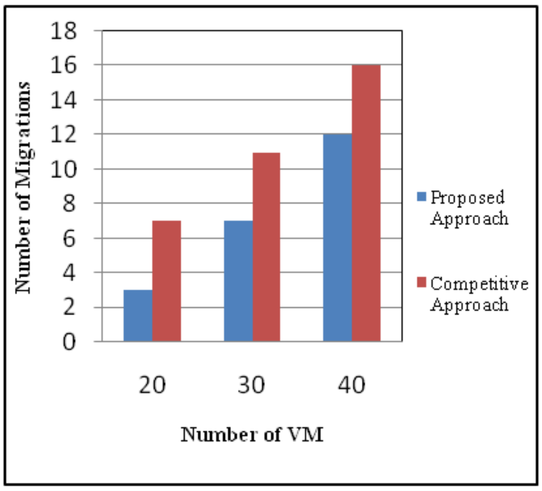

Figure 2: Number of Migrations

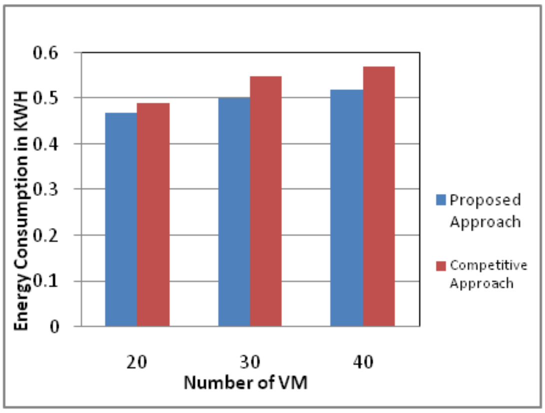

Figure 3: Energy Consumption

DOI 10.17148/IARJSET.2016.3322

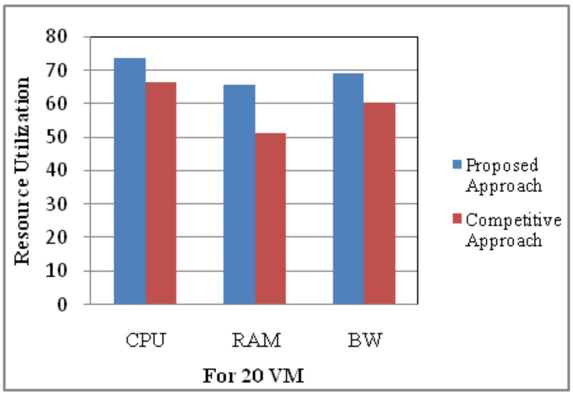

Figure 4: Resource Usage for 20 VM

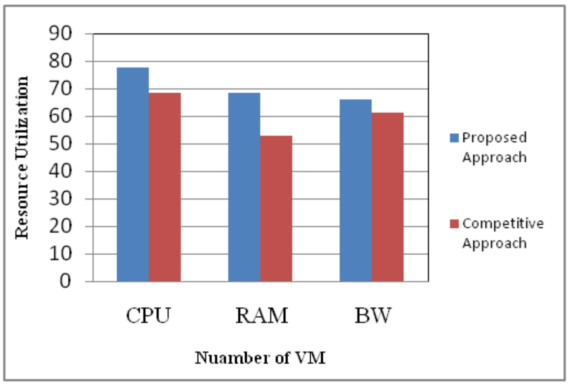

Figure 5: Resource Usage for 30 VM

Figure 2, 3, 4 and 5 says that proposed method for balancing the load gives improved results as compare to the base method. Reason to utilizing more CPU, memory and bandwidth is that our method minimizes the resource wastage.

VII. CONCLUSION

Virtualization is one of the core technologies in the cloud and enables the provider to deploy infrastructure services in cloud environment. Since a computing environment consist of several resources like CPU, RAM etc. So balance utilization or minimum resource wastage is the useful ways to enhance the performance of the physical machine. Load balancing in the cloud is the 3 steps procedure in cloud that is selecting PM source then selecting the VM for the migration and last selecting Pm as a target to place the VM. This paper present the solution for all steps involved in balancing. This paper introduced a method for balancing the load and minimizes the resource leakage. It is implemented in CloudSim simulator. Experiment results declare that our method gives the optimal result as compare to the base approach.

REFERENCES

[1] “Cloud computing,” [Online] available: http://en.wikipedia.org/wiki/ Cloud_computing, July 2014.

[2] R. K Gupta et al. “A Complete Theoretical Review on Virtual Machine Migration in Cloud Environment”, International Journal of Cloud Computing and Services Science (IJ-CLOSER), Vol.3, No.3, June 2014, pp. 172-178.

[3] R. K. Gupta et al.,” Survey on Virtual Machine Placement Techniques in Cloud Computing Environment”, International Journal on Cloud Computing: Services and Architecture (IJCCSA) ,Vol. 4, No. 4, August 2014, pp. 1 -7.

[4] R. Santhosh and T. Ravichandran, “A Survey on Cloud-Based Scheduling Algorithms”, International Journal of Communications and Engineering (IJCE), Volume 01– No.1, Issue: 01 May2013.

[5] S. K. Mandal and P. M. Khilar, “Efficient Virtual Machine Placement for On-Demand Access to Infrastructure Resources in Cloud Computing”, International Journal of Computer Applications (IJCA),Vol. 68, No.12, April 2013.

[6] “Hypervisor,” [Online] available: http://en.wikipedia.org/wiki/ hypervisor, June 2014

[7] “Xen, virtual machine manager in Cloud computing,” [online] available: http://www.xen.org, April 2013.

[8] R. Addawiyah et al., "Virtual Machine Migration Implementation in Load Balancing for Cloud Computing", six IEEE international conference, 2014.

[9] Ekta Gupta et al., “A Technique Based on Ant Colony Optimization for Load Balancing in Cloud Data Center”, proceeding of the 13th International Conference on Information Technology, December 2014, pp. 12-17.

[10] D. Gmach , “Resource pool management: Reactive versus proactive or let Ss be friends”. Computer Networks, 2009

[11] A. Beloglazov, R. Buyya, “Energy efficient allocation of virtual machines in cloud data centers”. 10th IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing ,2010.

[12] R. Calheiros, R Ranjan, César A. F. De Rose, R. Buyya, “ CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services” , 2011.

[13] M. Mishra, Anwesh Das, P. Kulkarni “Dynamic Resource Management using Virtual Machine Migration”, IEEE Magazine, September 2012.

[14] M. Mishra and A. Sahoo, "On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach", IEEE 4th International Conference on Cloud Computing, July 2011, pp. 275-282

DOI 10.17148/IARJSET.2016.3322