African Journal of Basic & Applied Sciences 7 (3): 169-175, 2015
ISSN 2079-2034
© IDOSI Publications, 2015
DOI: 10.5829/idosi.ajbas.2015.7.3.22303

# An Efficient Host Load Aware Algorithm for Scheduling Virtual Machines in a Cluster of Physical Machines in Datacenter

*[1]T. Thiruvenkadam and [2]V. Karthikeyani*

[1]Department of Computer Science, K.S.Rangasamy College of Arts and Science, Tiruchengode, India
[2]Department of Computer Science, Thiruvalluvar Govt., Arts College, Rasipuram, India

**Abstract:** Cloud computing is an emerging model that wish to share various resources and dispense services apparently among huge users of internet. IaaS is one of the services of cloud that can be considered as a pool of virtualized computer resources with deployment of software services. The Virtual Machine allocation problem in a cloud infrastructure is investigated by many researchers in the past. But the majority of the presented mechanisms paid no attention to the ever changing load of the physical host and dynamic nature of the Virtual Machine deployment requests that frequently reaches the cloud provider infrastructure. Here we present an efficient hybrid host load aware algorithm for scheduling virtual machines to a cluster of Physical hosts. We developed the algorithm based on two different methods, first by checking the load of the physical host, the load factor of a physical host can be measured by the way of analyzing utilization level of the individual resources like CPU, Memory and Network bandwidth. Second by considering the past utilization activities of a VM to a physical host. The proposed algorithm tries to enhance the rate of resource utilization through accommodating more number of virtual machines in a physical host at the same time without violating any SLA. It also reduces the power consumption by turning off the idle physical host on which no VMs are running.

**Key words:** Virtualization · Scheduling · Virtual Machine · Physical Machine Cluster · Load balancing

## INTRODUCTION

The cloud computing technology employs the Internet and remote servers to store and manage data and software applications. Cloud customers and organizations can make utilization of cloud applications without establishment and access their individual information at any machine with web access. Cloud computing considers a great deal more proficient and improved processing by pooled resources and data transmission. The term Cloud Computing refers to both the applications delivered as services over the Internet and the servers and system software in the data centers that provide those services. The term cloud computing got popular and the services offered by cloud IaaS, PaaS & SaaS became formalized.

### Cloud Service Model

**Software as a Service:** SaaS methods are accepted as an alternate for traditional software which exists in an individual computer or is distributed over a local area network. In SaaS no need for locally running software and Applications, delivered to users through a web browser, which are almost completely accumulated, administered and restructured in the cloud, SaaS runs on a Web browser, Gmail and Sales force are well-known products of SaaS. [1]The challenges of SaaS includes a lesser amount of facility to customize the application for particular business needs, complexity in integrating SaaS with existing software based infrastructure and budgeting suspicions inherent in pay-as you-go pricing models.

**Platform as a Service:** The end user of PaaS can deploy user created or obtained applications developed using programming languages, libraries, services and tools supported by the service provider on to the cloud.[2] using a PaaS, organizations can develop new applications rapidly and with high flexibility than with existing development platforms coupled directly to hardware resources. Running application development on a PaaS has a more number of advantages. Programmers and managers need not anxious about the infrastructure but the cloud provider holds all the care and maintenance of the underlying operating system(s), servers, storage and application containers.

**Corresponding Author:** Dr. V. Karthikeyani, Department of Computer Science, Thiruvalluvar Govt., Arts College, Rasipuram, India.

**Nfrastructure as a Service:** Infrastructure-as-a-Service (IaaS) is the most fundamental use of cloud computing. The virtualization technology is the base to form an IaaS platform. This proposes the entire computing resources for deploying and executing applications, storing data, or accommodating a company's complete computing environment. [3] The consumer has control over operating systems, storage and deployed applications but cannot manage or control the underlying cloud infrastructure. The service provider manages the virtualization, servers, networking and storage. This helps to keep away from expenses on hardware and diminish hazard of return on investment. Some of the most popular service providers of IaaS include Amazon, Microsoft, VMware, Rack space and Red Hat.

The confront with Iasi is includes lacuna in control over the computing hardware and network devices and disclosure to anonymous consumers who may be running virtual machines on the same physical server as your own organization's virtual machines. The additional clients in the same physical host can suck up network bandwidth and I/O which leads to performance overheads.

Virtualization technologies guarantee opportunities for cloud data centers to host applications on shared infrastructure. Data center expenses can be lessened by using virtual machines (Vs.) Cloud data center providers can create a huge number of virtual machines (VMs) for different types of workload and specification requirements.[4] Each VM is configured with a certain amount of computing resources which is adequate with workload requirements. The cloud service providers can consolidate all the VMs into a few numbers of physical hosts, keeping in mind the end goal to lessen the aggregate number of obliged physical servers and abusing server capacities all the more completely, permitting cloud providers to spare cash on equipment and vitality costs. VM consolidation method is the key sympathy toward attaining economy of scale in a cloud data center domain. [5]

**Related work:** Most of the Iasi cloud data centers uses virtualization technology since it provides a good flexibility in the provisioning and placement of servers and their associated workloads and cost savings [6, 7] while this model provides a number of advantages, it is essential to administer the allocation of virtual machines to the physical hosts in the data center. Even though a lot of researchers have been studied this virtual machine mapping problem in the past we draw attention to some of the closest work in perspective of our point.

In [8] the number of physical machines needed to deploy the requested virtual machine instances are reduced by combining time series forecasting techniques and bin packing heuristic but the model has not included the relationships between multiple resources, like CPU and I/O. In [9] the VM placement algorithms make use of the behavior of VMs to have some properties in general.

In [10] for the placement of virtual machines to physical machines a two level control management system is used and it uses combinatory and multi-phase efficiency to solve potentially inconsistent scheduling constraints. In [11], VM scheduling constraints are considered as single dimension in a multidimensional Knapsack problem.

In [12], the VM scheduling policy is primarily dealt out from the viewpoint of network traffic and three common scheduling algorithms have been introduced for Cloud computing and simulation results provided. In [13] the performing load balancing in data centers are intensively studied the heuristics has been used as a common approach among systems to enables the load balancing among physical servers. In [14] the performance variations have been identified and monitored in a physical server hosting VMs. A few simple VM placement algorithms like time-shared and space-shared were presented and compared in [15] and introduced a method to model and simulate Cloud computing environments, in which the algorithms can be implemented. In [16] pioneered methods for virtual machine migration and proposed some migration techniques and algorithms. [17] Evaluated most important load-balance scheduling algorithms for conventional Web servers. Vector Dot a novel load-balancing algorithm has been introduced in [18] to work with structured and multi-dimensional resources limitations by taking servers and storage of a Cloud into account. A countable measure of load imbalance on virtualized data center servers has been proposed in [19]. In [20] a comparative study of widely used VM placement strategies and algorithms for Cloud data centers has been presented. An overloaded resource based VM placement approach has been presented in [21]. In our previous study [22] the comparison of various VM scheduling algorithm has been presented and demonstrated the necessity of new efficient placement VM placement algorithm.

**Problem Formulation:** The major principle of the Iasi cloud computing system is that its user can make use of the resources to have good performance and economic benefits. With the support of virtualization innovation

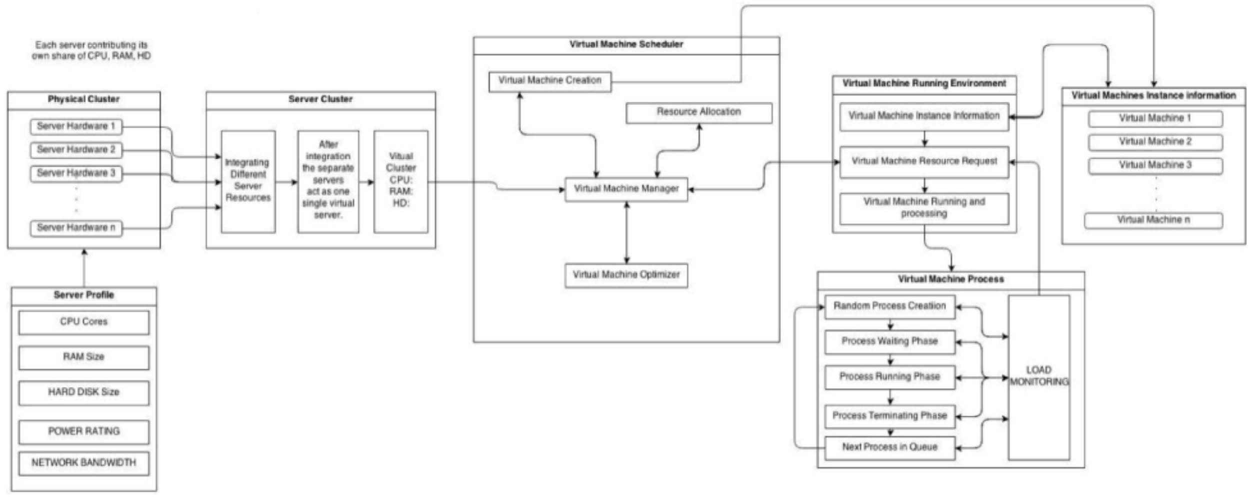
Each server contributing its own share of CPU, RAM, HD
Physical Cluster
Server Hardware 1
Server Hardware 2
Server Hardware 3
Server Hardware n
Server Profile
CPU Cores
RAM Size
HARD DISK Size
POWER RATING
NETWORK BANDWIDTH
Server Cluster
Integrating Different Server Resources
After integration the separate servers act as one single virtual server.
Vitual Cluster CPU: RAM: HD:
Virtual Machine Scheduler
Virtual Machine Creation
Resource Allocation
Virtual Machine Manager
Virtual Machine Optimizer
Virtual Machine Running Environment
Virtual Machine Instance Information
Virtual Machine Resource Request
Virtual Machine Running and processing
Virtual Machines Instance information
Virtual Machine 1
Virtual Machine 2
Virtual Machine 3
Virtual Machine n
Virtual Machine Process
Random Process Creation
Process Waiting Phase
Process Running Phase
Process Terminating Phase
Next Process in Queue
LOAD MONITORING

Fig. 1: Framework model for VM placement in a cluster of physical machines

the resources can be conveyed to the users in the form of virtual machines hence an efficient virtual machine allocation policy and management process is required to avoid underutilization or overutilization of the physical machines which may affect the quality of services of the Iasi cloud. The under utilization of servers is a well known expenditure concern in cloud management. Low utilization of server resources leads to the usage of more physical machines, increasing expenses for machine power and capital and operational expenses for cooling systems. Moreover, surplus machines require more carbon footprint. The overutilization of physical servers results in violating the SLA and quality of service constraints. Efficient allocation of Virtual machine instance request will meet client requirements, improve the resource utilization, increases the overall performance of the cloud computing environment and also decreases the number physical machines used. Therefore an efficient VM scheduling in Iasi is an important cloud computing problem to resolve.

**Description of Design Model:** To handle the VM scheduling problem we have proposed a physical host load aware scheduling algorithm and we have implemented this heuristics in JAVA using Net beans IDE. The below figure shows the framework model in which the proposed algorithm is implemented. Here the physical clusters can be formed by adding a set of physical servers each server contributing its own share of resources such as CPU cores, main memory, disk capacity and network bandwidth. The users can create virtual machine instances by giving their requirements for running the applications and the VM requests are submitted by the users to the computing system. As the submitted VMs enter to the cloud they are wait for their turn in the stack. The VM requests can be handled by the virtual machine scheduler and it finds the appropriate physical machine by estimating the VM size and checking for the availability and capacity of the physical machine when it finds the appropriate physical machine the VM scheduler immediately allocates the identified physical machine to the virtual machine instance request in queue and the required resource can be allocated to the virtual machine.

**Algorithm Design:** This is a simple and efficient method that uses the load factor of the physical machine and also input given by the user about the VM resource requirement. It also identifies the overloaded physical machine and selects the VM to migrate based on the past behavior of the VM and picks the appropriate PM based on its resource utilization rate. Then it discovers the underutilized PMs and migrates the VMs running on it to some other suitable PMs and turn it off in view of energy saving.

**Process of Virtual Machine Allocation:** Since accurately forecasting the resource requirement and behavior of the VM is not possible our algorithm utilizes the user deployed resource details of workload of the VM and considers the load factor of the physical machine as well as physical machine cluster to identify the appropriate PM for the given VM request. We use bin packing heuristic combined with three different algorithms to minimize the number of Physical machines required to place a set of VMs, quick and correct placement of VMs, maintain balanced load among the servers, increase the resource utilization rate and importantly doing all these things without violating any SLA agreements.

N number of virtual machines with resource requirements VR (CPU, Memory, N/W Bandwidth) to be placed on a set of M physical machines with resource capacities of PR(CPU, Memory, N/W Bandwidth) grouped in K number of physical machine cluster.

Consider *PM* as a set of all the physical machines in the entire system, where PM = {$PM_1$, $PM_2$, $PM_3$ … $PM_m$ }. m is the total number of the physical machines and an individual physical machine can be denoted as *PM x*, where *x* denote the physical machine number and range of x is (1 <= x <= N). Similarly, we have a set of VMs on each physical machine *x*, *VMx* = {VMx1, VMx2, VMx n} here n is the number of VMs on the physical server x. If we want to deploy VM y on the PMx then the load of the CPU, RAM and bandwidth has to be calculated individually.The CPU load of the PMx is denoted as follows

$$PMx(C,ts)=\sum_{j=1}^{n} VMxj(C,ts) \tag{1}$$

The amount of RAM utilized by all the VMs of PMx at the time interval ts can be denoted as follows,

$$Mx(R,ts)=\sum_{j=1}^{n} VMxj(R,ts) \tag{2}$$

The amount of network bandwidth utilized by all the VMs of PMx at the time interval ts can be denoted as follows

$$PMx(NB,ts)=\sum_{j=1}^{n} VMxj(NB,ts) \tag{3}$$

Where PMx represents the $x^{th}$ physical machine of the Physical Machine Cluster k, VMxj represents $j^{th}$ virtual machine of the PMx and C, R and NB denotes the amount of CPU, RAM and bandwidth utilized by all the VMs of the PMx respectively.

Hence derived from (1),(2) and (3) the weighted average load of the Physical Machine Cluster k at time interval ts can be denoted as follows

$$PMx(WL,ts)=\sum_{j=1}^{n} VMxi(WL,ts) \tag{4}$$

Where PMCk represents the $k^{th}$ physical machine cluster of the datacenter,WL represents the weighted load of physical machine cluster at time interval ts and PMx represents the $x^{th}$ physical machine of the Physical Machine Cluster k.

At any time interval the total VM load of a PM should not exceed the host capacity

$$\sum_{resource} PMxW_{resource\,usage}(ts) \leq THvalue \leq \sum_{resource} PMxW_{resource\,capacity} \tag{5}$$

Where resource _ {CPU, RAM, Network Bandwidth} and $W_{resource}$ is the weight associated with each resource TH value is the threshold value set by the administrator if the load goes beyond this value the host can be considered as overloaded host and the selected VMs has to be migrated to other appropriate physical machines.

**Dynamic IM Placement:** In this process the objective is to place the VMs in PMs in a way that the total number of PMs required to place all the VMs is decreased. So we considered this a multi potential bin packing problem since this is a NP-hard problem, we provide a heuristic based on multiple policy. In the earlier stages of allocation most of the PMs are underutilized or not used so our heuristics works as like the first fit scheduler which is a simplest one to implement and which increases the response time of VM placement. As the number of VM grows in the datacenter the utilization level of PM is also being considered by our heuristic which really helps in maintaining the balanced load among servers. Towards the closing stages the heuristic works according to the nature of the VMs workload that is gathered from the user provided hints which helps in avoiding the bottleneck of a particular resource as well as avoiding the violence of any SLA agreements. The algorithm which is used to achieve these things is given below.

**Algorithm 1:** Dynamic VM placement

Step 1: The VM requests given by the user at the time ti is considered for allocation and scans the values of no CPU cores, amount of RAM and amount of N/W bandwidth required.

Step 2: In this algorithm the scheduler maintains an index table for physical clusters and physical machines as well as their states whether available or busy.

Step 3: The scheduler scans the index table of the physical cluster for the load below 50 %, from top until the first available physical cluster is found or the index table is scanned fully.
Step 4: If the physical cluster is found then scan the index table of physical machines for the load below 50 % in all three major resources, from the top until the first physical machine is found.
Step 5: When found return the ID of the physical machine to the main controller
Step 6: Assign the VM to the identified PM.
Step 7: Update the index table of the PM and Physical cluster.
Step 9: Go to the step 1
Step 8: If not found then scheduler scans the index table of the physical cluster for the load below 70 %, from top until the first available physical cluster is found or the index table is scanned fully.
Step 9: If the physical cluster is found scan the index table of the PMs based on the requirements of the requested VM.
Step 10: If the requested VM is a CPU intensive then scan the PM index table for the amount of CPU utilized is below 70 %, from the top until the first physical machine is found.
Step 11: When found return the ID of the physical machine to the main controller
Step 12: Assign the VM to the identified PM.
Step 13: Update the index table of the PM and Physical cluster and go to the step 1
Step 14: If the requested VM is a memory intensive then scan the PM index table for the amount of RAM utilized is below 70%, from the top until the first physical machine is found.
Step 15: When found return the ID of the physical machine to the main controller
Step 16: Assign the VM to the identified PM.
Step 17: Update the index table of the PM and Physical cluster and go to the step 1
Step 18: If the requested VM is a network intensive then scan the PM index table for the amount of network bandwidth utilized is below 70%, from the top until the first physical machine is found.
Step 19: When found return the ID of the physical machine to the main controller
Step 20: Assign the VM to the identified PM.
Step 21: Update the index table of the PM and Physical cluster and go to the step 1
Step 22: If Physical Cluster is not found. The scheduler scans the index table for the load below 80 %, from top until the first available physical cluster is found or the index table is scanned fully
Step 23: If found scan the index table of the PMs based on the requirement of the requested VM.
Step 24: If the requested VM is a CPU intensive then scan the PM index table for the least number of CPU cores utilized from the top until the first physical machine is found.
Step 25: If found check the host has enough CPU cores to fulfill the VMs CPU requirement and will not surpass 90% of load after placing the new VM, then return the ID of the physical machine to the main controller.
Step 26: Assign the VM to the identified PM.
Step 27: Update the index table of the PM and Physical cluster and go to the step 1.
Step 28: Else go to step 22
Step 29: If the requested VM is a memory intensive then scan the PM index table for the least amount of RAM utilized from the top until the first physical machine is found.
Step 30: If host has enough RAM to fulfill the VMs memory requirement and will not surpass 90% of load after placing the new VM, then return the ID of the physical machine to the main controller.
Step 31: Assign the VM to the identified PM.
Step 32: Update the index table of the PM and Physical cluster and go to the step 1.
Step 33: Else go to step 22
Step 34: If the requested VM is a network intensive then scan the PM index table for the least amount of network bandwidth utilized from the top until the first physical machine is found.
Step 35: If host has enough bandwidth to fulfill the VMs bandwidth requirement and will not surpass 90% of load after placing the new VM, then return the ID of the physical machine to the main controller.
Step 36: Assign the VM to the identified PM.
Step 37: Update the index table of the PM and Physical cluster and go to the step 1.
Step 38: Else go to step 22

**Load Balancing among Physical Servers:** Since virtual machine workloads frequently change eventually, the well primary placement choices is not sufficient to maintain

Table 1: Properties required for the index table of physical machine and physical machine cluster

| S.No | Physical Machine | Physical Machine Cluster |
|---|---|---|
| 1 | To Total number of VMs placed | Total number of PMs |
| 2 | To Total number of VMs in each type (CPU intensive, RAM intensive, N/W intensive) | Total number of PMs exhausted |
| 3 | The percentage of load of the PM in each resource type individually | The cumulative percentage of the load of the entire PMs |
| 4 | Total number of CPU cores utilized and available | The list of PMs which can be used to place the CPU intensive VMs |
| 5 | Total amount of RAM utilized and available | The list of PMs which can be used to place the memory intensive VMs |
| 6 | Amount of n/w bandwidth utilized and available | The list of Pms which can be used to place the N/W Bandwidth intensive Vms |

the load balance. So it is essential to dynamically rework placements to make QOS constraints are to be satisfied while change in the data center load.

Based on the information provided by the monitor drivers Whenever the load factor of the host goes beyond the threshold value set by the administrator, the below algorithm triggers and migrates VMs to other suitable hosts.

**ALGORITHM 2:** Load Balancing among Physical Servers

Step 1: If any PM resource utilization surpasses the threshold value that can be considered as an overloaded host
Step 2: Search for the VMs which resource utilization has gone beyond its allocated resource limit on the overloaded host and list all those VMs based on their resource utilization level.
Step 3: Migrate the first VM in the list to the next suitable PM based on the algorithm 1.
Step 4: Check the PM load against threshold value.
Step 5: If the load is below threshold value then exit. Else repeat step 3 until condition satisfies.

**Vm Consolidation:** The thought here is to recognize PM running at low use and afterward relocating VMs inhabitant on those PMs to different PMs such that those low use PMs can be taken logged off. A straightforward algorithm to attain this is displayed in algorithm 3.

**ALGORITHM 3:** Vm Consolidation to Avoid Low Utilization of Servers

Step 1: The scheduler evaluates a ranking policy for the PMs using the information gathered by the monitor drivers.
Step 2: PM with the highest load ranked top and check the load factor of the least ranked PM is below 20%.
Step 3: Select and migrate all the VMs individually to the next suitable PMs based on the algorithm 1.
Step 4: If there is no more VM in the list then remove the PM from the list and power off.

## CONCLUSION

We presented our algorithm which will address the problem of mapping the Vms into PMs such that the number physical host used is minimized, the overutilization and underutilization of the resources of a host can be identified and resolved at the same time without violating any SLA agreements. Since we consider this as a multi potential bin packing problem we combined three different heuristics which considers load factor of hosts along with user provided information at the various stages of placing the VMs in physical hosts. We believe that this load aware algorithm really optimizes the VM placement.

The presented algorithm is implemented in JAVA Netbeans IDE. Then we planned to integrate the algorithm with CloudSim simulator for simulation to assess the execution and performance of our heuristics with some of the existing scheduling algorithm in terms of Response Time, Load Balancing among servers, Reasonable Resource Utilization, energy consumption, Minimum number of active PMs and Higher profit.

## REFERENCE

1. Chen, K. and W.M. Zheng, 2009. Cloud Computing: System instance and Current State, Journal of Software, 20(5): 1337-1348.
2. Xu, Z.W. and H.M. Liao, 2008. The Classification research of Network Computing System, Journal of Computing Machine, 18(9): 1509-1515.
3. Zhang, G.W., R. He and Y. Liu, 2008. The Evolution based on Cloud Model, Journal of Computing Machine, 7: 1233-1239.

4. Breitgand, D. and A. Epstein, 2011. Sla-aware placement of multivirtual machine elastic services in compute clouds. In Integrated Network Management (IM), 11 IFIP/IEEE International Symposium on, pp: 161-168.
5. Xiaoqiao Meng, Canturk Isci, Jeffrey Kephart, Li Zhang, Eric Bouillet and Dimitrios Pendarakis, 2010. Efficient resource provisioning in compute clouds via vm multiplexing. In Proceeding of the 7th international conference on Autonomic computing, New York, NY, USA, pp: 11-20.
6. Hewlett Packard Web Site, Introducing Integrity Virtual Machines white paper, http://docs.hp.com/en/9987/Intro VM 2.1.pdf.
7. VMWareWeb Site, Server Consolidation and Containment Solutions Brief, http://www.vmware.com/pdf/server consolidation.pdf.
8. Norman Bobroff andrzej Kochut and Kirk Beaty, Dynamic Placement of Virtual Machines for Managing SLA Violations.
9. Sindelar, M., R.K. Sitaraman and P. Shenoy, 2011 Sharing-Aware Algorithms for Virtual Machine Colocation, In: Proceedings of the 23rd ACM Symposium on Parallelismin Algorithms and Architectures, San Jose, California, USA.
10. Xu, J. and J.A.B. Fortes, 2010. Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. In: Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing. Hangshou, PR of China.
11. Singh, A., M. Korupolu and D. Mohapatra, 2008. Server-Storage Virtualization: Integration and Load Balancing in Data Centers. In: Proc. of the 2008 ACM/IEEE conference on Supercomputing (SC'08), Austin, TX, 53:1-53:12.
12. Meng, X., V. Pappas and L. Zhang, 2010. Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement, In: Proceedings of IEEE INFOCOM. San Diego, CA, USA.
13. Kumar, S., V. Talwar, V. Kumar, P. Ranganathan and K. Schwan, 2009. Manage: Loosely Coupled Platform and Virtualization Management in Data Centers. In: Proceedings of the 6th international conference on Autonomic computing. ACM, Barcelona, Spain, pp: 127-136.
14. Wood, T., P. Shenoy, A. Venkataramani and M. Yousif, 2007. Black-box and Gray-box Strategies for Virtual Machine Migration. In: Proc of the 4th USENIX Symposium on Networked Systems Design and Implementation. Cambridge, MA.
15. Bobroff, N., A. Kochut and K. Beaty, 2007. Dynamic Placement of Virtual Machines for Managing SLA Violations. In: Proc of the 10th IFIP/IEEE International Symposium on Integrated Network Management. Munich, Germany.
16. Wood, T., 2007. Black-box and gray-box strategies for virtual machine migration, proceedings of Symp. on Networked Systems Design and Implementation (NSDI).
17. Zheng, H., L. Zhou and J. Wu, 2006. Design and implementation of load balancing in web server cluster system, Journal of Nanjing University of Aeronautics & Astronautics, 38(3): 00-00.
18. Singh, A., M. Korupolu and D. Mohapatra, 2008. Server-storage virtualization: Integration and load balancing in data centers, Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, pp: 1-12.
19. Arzuaga, E. and D.R. Kaeli, 2010. Quantifying load imbalance on virtualized enterprise servers, Proceedings of WOSP/SIPEW'10, San Jose, California, USA, pp: 28-30.
20. Tian, W., C. Jing and J. Hu, 2011. Analysis of resource allocation and scheduling policies in cloud datacenter, Proceedings of the IEEE 3rd International Conference on Networks Security Wireless Communications and Trusted Computing.
21. Thiruvenkadam, T. and Dr. V. Karthikeyani, 2014. An approach to virtual machine placemnt problem in a datacenter environment based on overloaded resource, International Journal of Computer Science and Mobile Computing, 3(6): 837-842.
22. Thiruvenkadam, T. and Dr. V. Karthikeyani, 2014. A Comparative study of VM Placement Algorithms in Cloud Computing Environment, In proccedings if 07th SARC-IRF International Conference, India.