# Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

OPEN ACCESS Research Journal of Information Technology

ISSN 1815-7432
DOI: 10.3923/rjit.2016.98.104

# Research Article
# Tasks Scheduling in Cloud Computing Environment using Workflowsim

[1]Pradeep Singh Rawat, [2]Priti Dimri and [3]G.P. Saroha

[1]Department of Computer Science and Engineering, Faculty of Computer Science and Engineering, DIT University, Dehradun, India
[2]Department of Compuer Application, Faculty of Computer Applicaton, G.B. Pant Engineering College, Ghurdauri Pauri, India
[3]University Computer Center, Maharshi Dayanand University, Rohtak, India

## Abstract
**Background:** In the era of computing technology revolution is continuously changing. Cloud computing is at the top of the stack to pop the computation paradigm. For real deployment of application scheduling is the key challenge in cloud computing. **Methodology:** To deal with this challenge researcher need to utilize the modeling and simulation features of cloud simulation tools. **Results:** The research study primarily focused on to identify the best scheduling algorithm for real deployment of application using simulation results. Research study has focused on different scheduling policies using workflowsim and identify the best policy for constant configuration. **Conclusion:** Simulation results are obtained by using 6 different scheduling algorithms with constant configuration at virtual machine, host and datacenter level.

Key words: MIPS, IaaS, PaaS, VM, MINMIN, MAXMIN, FCFS, MCT

Received: April 01, 2016 Accepted: May 17, 2016 Published: September 15, 2016

Citation: Pradeep Singh Rawat, Priti Dimri and G.P. Saroha, 2016. Tasks scheduling in cloud computing environment using workflowsim. Res. J. Inform. Technol., 8: 98-104.

Corresponding Author: Pradeep Singh Rawat, Department of Computer Science and Engineering, Faculty of Computer Science and Engineering, DIT University, Dehradun, India



Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

## INTRODUCTION

Distributed computing leads to utility base computing. The concept of utility computing leads to a new technology called cloud computing and grid computing. All computing paradigms are used by both academia and industry to store and retrieve the files and necessary documents. Files stored at remote location need to be accessed according to the service level agreement between users and cloud service providers. The main issue is to schedule the incoming requests in an efficient way with minimum response time and resource optimization. Key benefits of cloud computing i.e., performance, productivity, cost reduction and location and device independence. Concern issues in cloud computing is security and load balancing. This study highlights the load balancing concern. Different load balancing algorithms like FCFS, round robin, active-VM monitoring and throttled are used for executing clients request with a minimum response time and also assigning the requests to the available virtual machines[1]. Load balancing is to equalize the load among the node inside the datacenters to control the performance evaluation parameters. Datacenters are geographically distributed so the constraints such as high communication delays, underutilization of the resources are difficult to address clearly and efficiently. Performance evaluation matrices include response time, datacenters processing time, communication delay and throughput. Many of the resources do not participate in executing the requests and hence leads to imbalance of cloud system. Load balancing is the key challenge in cloud computing environment. It is essential at service provider's end that every virtual machine in the cloud system should do the same amount of work to minimizing the response time and finish time of the cloudlets running over virtual machines. This study presents a novel simulation configuration for VM-assignment to the cloudlet to identify the best load balancing algorithm. Classification of load balancing algorithm is based on environment or system load[2,3]. It can be helpful to allocate the incoming jobs to available virtual machines. Here, the virtual machine assigned depending on its status i.e., over loaded or under loaded. Research study highlights the static load balancing algorithms. This study shows the simulation configuration to create one data centers and 5 virtual machines, 1 host and bandwidth inside the datacenter is 15 MBs. For minimum finish time of cloudlets scheduling on virtual machine six different scheduling algorithms are used.

Figure 1 shows the basic architecture used for modeling and simulation of cloud computing environment using cloudsim[4]. Simulation architecture shown in Fig. 1 is used for experimental simulation of cloud environment. To provide the virtualization 5 virtual machines are used with single host. Architecture includes all the components required to prepare for simulation. At IaaS (Infrastructure as a Service) level one host is added with fixed MIPS (million instructions per seconds) rating. To run the cloudlet or cloud tasks PaaS (platform as a service) capability model uses 5 virtual machines.

Figure 2 shows the abstract view of virtual cloud simulation architecture. It includes the cloud main resource

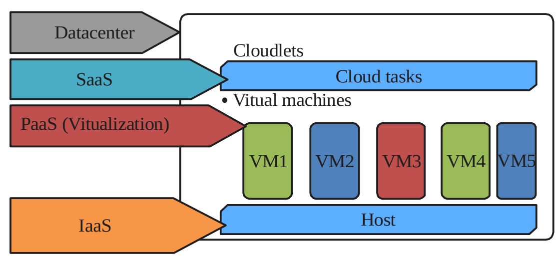


Fig. 1: Layered architecture for simulation of cloud computing environment in cloudsim

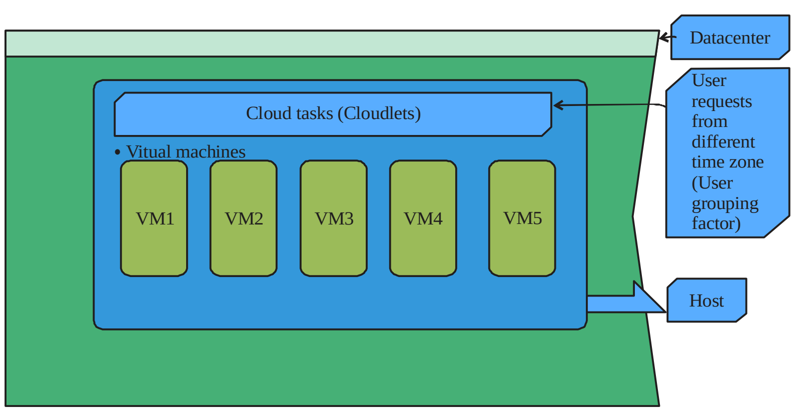


Fig. 2: Abstract view of cloud resources

i.e., datacenter. Inside datacenter host and virtualization is created with geographically distributed servers across the globe. User grouping factor and request grouping factors are associated with the cluster base computing paradigm. This architecture can also used for simulation in graphical mode so that user can set the parameters at main configuration level. For this purpose cloudsim based tool i.e., cloud analyst is the best choice for simulation[5].

## MATERIALS AND METHODS

To identify the best technique for scheduling the tasks on virtual machine quantitative method is used. For real deployment of application simulation plays an important role. Input parameters shown in Table 1 for capability model PaaS. Simulation environment first of all cloud users are setup which are directly proportional to number of data center broker. The CIS is initialized. Then cloud main resource datacenter is created with host configuration. This provides the abstract view of cloud main resource. Simulation parameters are set then simulation started by calling the main function start simulation inside the cloudsim class. The basic approach used for the research study generates the data in quantitative forms. The quantitative approach focuses on inferential and experimental and simulation approaches. This study highlight with artificial environment with which some relevant information and data is generated for real deployment of applications.

Table 1 shows that the 5 virtual machines are used having same configuration. The MIPS rating is provided by single processing element in timeshared mode. The MIPS rating at virtual machine level is limited by host MIPS at infrastructure level.

Table 2 shows that the host properties are set to provide the capability IaaS model. Storage, computing and network resources at host levels are providing the infrastructure parameters for service level agreement between user and cloud service providers.

Table 3 shows the characteristics of cloud main resource i.e., datacenter. Datacenter is the heart of cloud computing environment which provides the infinite resources to the cloud user for service level agreement. Time zone 10.0 indicates the geographical location of datacenter across the globe. Within the datacenter to minimize the communication delay bandwidth parameter is allocated i.e., 15 Mbps.

Table 1: Virtual Machine (VM) parameters

| VM ID | Image size | RAM | MIPS | BW | PES No. | VMM |
|---|---|---|---|---|---|---|
| 1 | 10000 | 512 | 100 | 1000 | 1 | Xen |
| 2 | 10000 | 512 | 100 | 1000 | 1 | Xen |
| 3 | 10000 | 512 | 100 | 1000 | 1 | Xen |
| 4 | 10000 | 512 | 100 | 1000 | 1 | Xen |
| 5 | 10000 | 512 | 100 | 1000 | 1 | Xen |

Table 2: Host configuration

| Host ID | MIPS | RAM | Storage | BW |
|---|---|---|---|---|
| 0 | 2000 | 2048 | 100000 | 10000 |

Table 3: Datacenter characteristics

| | |
|---|---|
| System architecture | "x86" |
| Operating system | "Linux" |
| VMM | "Xen" |
| Time_zone time zone this resource located | 10.0 |
| Cost (The cost of using processing in this resource) | 3.0 |
| Cost per Mem (The cost of using memory in this resource) | 0.05 |
| Cosupt per storage (The cost of using storage in this resource) | 0.1 |
| Cost per BW (The cost of using BW in this resource) | 0.1 |
| Max transfer rate (The bandwidth within a datacenter in MBs) | 15 Mbps |

## RESULTS

Table 4 shows the results for the configuration parameters. Simulation results are corresponding to the different scheduling algorithms with number of datacenter = 1, No. of host = 1 and No. of virtual machine = 5. Scheduling algorithms provide the different time span for different virtual machine. Simulation results shows that MCT scheduling provides the minimum finish time (msec). For real deployment of application with configuration our priority should be MCT algorithm for tasks scheduling across virtual machine.

Figure 3 shows the variation of finish time of tasks across the different virtual machine using MINMIN scheduling policy. While preparing for simulation number of virtual machine = 5 are allocated to the tasks. Virtual machine with ID = 4 provides the minimum finish time. Virtual machine 3 takes maximum time due to over loading. Performance of virtual machine level can be improved by using effective task scheduling.

Figure 4 shows the variation of finish time of tasks across the different virtual machine using MAXMIN scheduling policy. While preparing for simulation number of virtual machine = 5 are allocated to the tasks. Virtual machine with ID = 4 provides the minimum finish time. Virtual machine ID = 2 takes maximum time due to over loading. Performance of virtual machine level can be improved by using effective task scheduling.

Figure 5 shows the variation of finish time of tasks across the different virtual machine using MCT task scheduling policy. While, preparing for simulation number of virtual machine = 5 are allocated to the tasks. Virtual machine with ID = 1 provides the minimum finish time.

Table 4: Comparison of different scheduling algorithms

| Scheduling algorithm | No. of datacenter | No. of host | No. of virtual machine | Virtual machine ID | Time span/finish time (msec) |
|---|---|---|---|---|---|
| FCFS | 1 | 1 | 5 | 1 | 1.20 |
| | | | | 1 | 1.34 |
| | | | | 2 | 1.41 |
| MINMIN (minimum-minimum) | | | | 3 | 1.81 |
| | | | | 4 | 1.37 |
| | | | | 5 | 1.60 |
| | | | | 1 | 1.42 |
| | | | | 2 | 1.66 |
| MAXMIN (maximum-maximum) | | | | 3 | 1.44 |
| | | | | 4 | 1.32 |
| | | | | 5 | 1.43 |
| | | | | 1 | 3.09 |
| Round robin | | | | 1 | 1.34 |
| | | | | 2 | 1.36 |
| MCT (Minimum completion time) | | | | 3 | 2.09 |
| | | | | 4 | 1.41 |
| | | | | 5 | 1.35 |
| Data aware | | | | 1 | 2.11 |

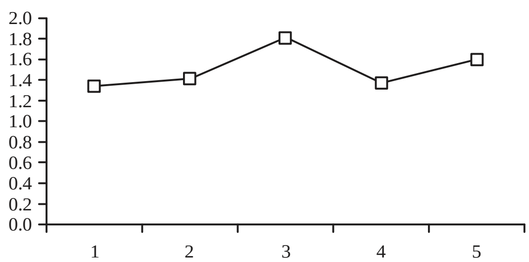


Fig. 3: Virtual machine ID versus finish time of cloudlet with MINMIN algorithm

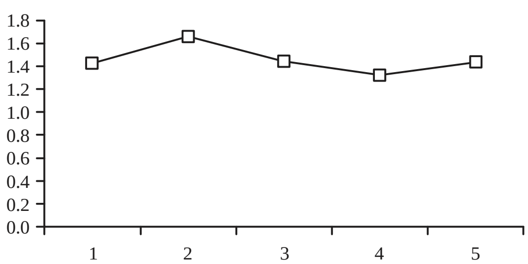


Fig. 4: Virtual machine ID versus finish time of cloudlet with MAXMIN algorithm

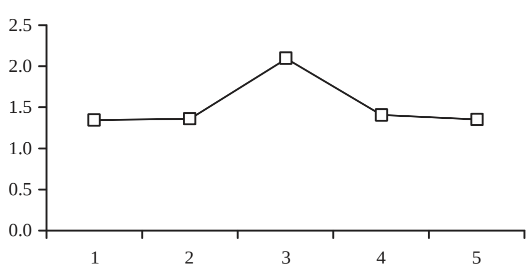


Fig. 5: Virtual machine ID versus finish time of cloudlet with MCT algorithm

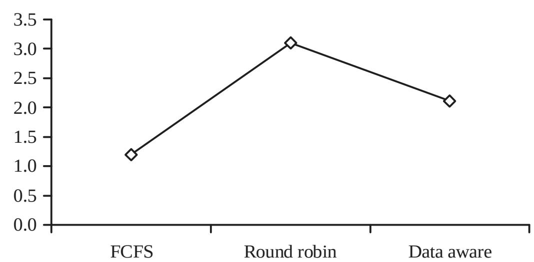


Fig. 6: Virtual machine ID versus finish time (msec) of cloudlet with 3 scheduling algorithm

Virtual machine 3 takes maximum time due to over loading. Performance of virtual machine level can be improved by using effective task scheduling.

Figure 6 shows the variation of finish time of tasks across the different virtual machine. Three task scheduling policies i.e., First Come First Serve (FCFS), round robin and data aware simulation results are compared for same input parameters. While preparing for simulation number of virtual machine = 5 are allocated to the tasks. Task scheduling FCFS provides the minimum finish time. Task scheduling policy data aware provides the maximum finish time due to over loading. Performance of virtual machine level can be improved by using effective task scheduling policy for resource optimization.

Figure 7 shows the variation of finish time of tasks across the different virtual machine using 3 optimal task scheduling policy. While preparing for simulation number of virtual machine = 5 are allocated to the tasks. Virtual machine with ID = 1, 2, 3, 4, 5 provides the best performance in case different task scheduling algorithm. Performance of virtual machine level can be improved by using effective task scheduling. Simulation results are obtained with fixed

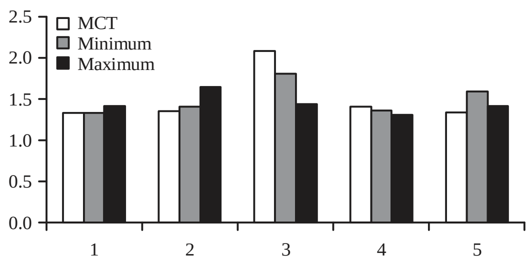


Fig. 7: Comparison of three scheduling algorithms with virtual machine ID versus finish time of cloudlet

configuration i.e., No. of datacenter = 1, No. of host = 1, No. of virtual machine = 5. In case of virtual machine with ID = 1 MINMIN and MAXMIN scheduling algorithm provides the approximately same task finish time (msec). When task is allocated to virtual machine ID = 2 then MCT is best choice for real deployment of application. In similar way the status of virtual machine is identified for cloudlet allocation. For virtual machine 3, 4 MAXMIN scheduling is right choice and in case of virtual machine with ID = 5 MCT scheduling is used for resource optimization with minimum finish time.

## DISCUSSION

The key challenge of cloud computing paradigm is considered by the researcher. Load balancing and security is one of the key issue in cloud computing. This research study highlights the different class of scheduling algorithms. Before highlighting the actual research findings previous research study is needed to discuss in favor as well as in contradiction. The cloud facility is a network of geographically distributed datacenters across the globe. Each datacenter consist hundreds of storage and computing servers. When a user submits a task (popularly known as cloudlet) it is handled by the datacenter controller which is directly associated with VM load balancer. Minimum scheduling algorithm[2], it starts with a set of tasks. Then the resource which has the minimum completion time for all tasks is found. Next, the task with the minimum size is selected and assigned to the corresponding resource (hence the name Min-Min). Finally, the task is removed from set and the same procedure is repeated by Min-Min until all tasks are assigned. The method is simple but it does not consider the existing load on a resource before assigning a task. So proper load balance is not achieved. Load balance improved Min-Min scheduling algorithm (LBIMM)[2]. It starts by executing Min-Min algorithm at the first step. At the second step it chooses the smallest size task from the heaviest loaded resource and calculates the completion time for that task on all other resources. Then the minimum completion time of that task is compared with the makespan produced by Min-Min. If it is less than makespan then the task is reassigned to the resource that produce it and the ready time of both resources are updated. The process repeats until no other resources can produce less completion time for the smallest task on the heavy loaded resource than the makespan. Thus the overloaded resources are freed and the under loaded or idle resources are more utilized.

This makes LBIMM to produce a schedule which improves load balancing and also reduces the overall completion time. But still it does not consider priority of a job while scheduling. User-priority awared load balance improved Min-Min scheduling algorithm (PA-LBIMM)[2]. User priority is incorporated with LBIMM algorithm to develop PA-LBIMM. This algorithm will first divide all the tasks into two groups G1 and G2. The G1 is for the VIP users' tasks having higher priority requirement. The G2 is for the ordinary users' tasks. The higher priority tasks in G1 are scheduled first using the Min-Min algorithm to assign the tasks to the VIP qualified resources set. Then the tasks with lower priority are scheduled to assign them to all the resources by Min-Min algorithm. At the end, the load balancing function is processed to optimize the load of all resources to produce the final schedule. The algorithm is only concerned with the makespan, load balancing and user-priority. It does not consider the deadline of each task. The Max-Min algorithm, it works as the Min-Min algorithm. But it gives more priority to the larger tasks. The jobs that have large execution time or large completion time are executed first. The problem is that smaller jobs have to wait for long time. Opportunistic Load Balancing (OLB). The OLB is a static load balancing algorithm whose goal is to keep each node in the cloud busy so does not consider the current load on each node. It attempts to dispatch the selected job to a randomly selected available VM[6]. The datacenter controller[5], uses a VM load balancer to determine which VM should be assigned the next request for processing. The VM load balancer can use different algorithms for load balancing. To achieve the resource optimization and cost minimization simulation plays an important role. The study highlights the scheduling algorithm MINMIN for task scheduling. However, OLB does not consider the execution time of the task in that node. This may cause the task to be processed in a slower manner increasing the whole completion time (make span) and will cause some bottlenecks since requests might be pending waiting for nodes to be free[7]. Mahalle *et al.*[7] have developed active monitoring load balancer algorithm which maintains information about each VMs and the number of requests currently allocated to which VM. When a request to allocate a new VM arrives, it identifies the least loaded VM. If there are more than one, the first identified is selected. Active VM load balancer returns the VM id to the data center controller the datacenter controller sends the request to the VM identified by that ID. Datacenter controller notifies the active VM load balancer of the new allocation. Domanal and Reddy[8] have developed modified throttled algorithm which maintains an index table of virtual machines and also the state of VMs similar to the throttled algorithm. There has been an attempt made to improve the response time and achieve efficient usage of available virtual machines. Proposed algorithm employs a method for selecting a VM for processing client's request where, VM at first index is initially selected depending upon the state of the VM. If the VM is available, it is assigned with the request and id of VM is returned to datacenter, else -1 is returned. When the next request arrives, the VM at index next to already assigned VM is chosen depending on the state of VM and follows the above step, unlikely of the throttled algorithm, where the index table is parsed from the first index every time the datacenter queries load balancer for allocation of VM. B. Wickremasinghe, R.N. Calheiros and Rajkumar Buyya have developed throttled algorithm which is completely based on virtual machine. Here, the client first requests the load balancer to check the right virtual machine which access that load easily and perform the operations which is given by the client. In this algorithm the client first requests the load balancer to find a suitable virtual machine to perform the required operation In the present study it is considered active-VM load balancer and proposed VM-assign algorithm for comparison. Our main focus is to distribute the load efficiently on the available virtual machines and ensuring that under or over utilization of the resources/virtual machines will not occur in the cloud system.

Load Balance Min-Min (LBMM)[9], this method uses Min-Min scheduling algorithm as its base. It uses a three level hierarchical framework. Request manager which is in the first level of the architecture is responsible for receiving the task and assigning it to one service manager in the second level of LBMM. After receiving the request, service manager divides it into subtasks to speed up the processing. Then service manager assigns the subtask to a service node for execution based on different attributes such as the remaining CPU space (node availability), remaining memory and the transmission rate. This algorithm improves the load unbalance of Min-Min and minimizes the execution time of each node but does not specify how to select a node for a complicated task requiring large-scale computation. The 2-phase load balancing algorithm[9] proposed this algorithm combining OLB and LBMM to have a better execution time and to balance the load more efficiently. A queue is used to store tasks that need to be carried out by manager. In the first phase OLB scheduling manager is used to assign job to the service manager. In the second phase LBMM algorithm is used to choose the suitable service node to execute the subtask by the service manager. The problem associated with this approach is that it applicable only in static environment. Honey bee Foraging algorithm[10], it is a decentralized honeybee-based nature-inspired load balancing technique for self-organization. It achieves global load balancing through local server action[11]. This algorithm is derived from the behavior of honey bees for foraging and harvesting food. Forager bees search for food sources and after finding advertise this using waggle dance to present quality of nectar or distance of food source from hive[12]. Harvester bees then follow the foragers to the location of food to harvest it. While using virtual machine to run the cloudlets scheduling policy is implemented for resource optimization. Simulation results depends on cloud environment which includes the main configuration, advanced configuration and internet characteristics parameters i.e., bandwidth and delay among different pair of geographic regions. The simulation results shows virtual machine 1, 2, 5 MCT algorithm provides the minimum finish time for cloudlet run and.

## CONCLUSION

In the era of cloud computing task scheduling and security are the key challenges for cloud users. User need to ensure the service level agreement before shifting towards the cloud service providers. This research study focus on task scheduling challenge. For resource optimization and cost minimization user should go through the simulation results for real deployment. Simulation results are obtained by using cloud simulation tool which are helpful to identify the best configuration for their organization. In this research study it is identified that MCT task scheduling algorithms is best for task scheduling across virtual machines. This provides the minimum finish time for cloudlet run. For service level agreement with real cloud user should follow the MCT scheduling algorithm while deploying the cloudlet on virtual machine with ID = 1, 2, 5. The MAXMIN scheduling is the right choice for deployment of cloud tasks on virtual machine with ID = 3, 4. This study highlights cloudlet finish time (msec) is the evaluation matrices for resource optimization in real cloud.

## ACKNOWLEDGMENTS

Weiwei Chen; Deelman are wished to thank you for implementing the simulation tool workflowsim. This study is partially supported by Professor G.P. Saroha Director MDU Rohatak haryana, Dr. Priti Dimri Associate professor G.B. Pant Engineering College Pauri Garhwal. This study was primarily carried out under the supervision of guide and co-guide. It now continues as a joint effort between mentor and researchers motivation. This study substantially focuses on Workflowsim base simulation for real deployment and resource optimization.

## REFERENCES

1. Al Nuaimi, K., N. Mohamed, M. Al Nuaimi and J. Al-Jaroodi, 2012. A survey of load balancing in cloud computing: Challenges and algorithms. Proceedings of the IEEE 2nd Symposium on Network Cloud Computing and Applications, December 3-4, 2012, London, UK., pp: 137-142.
2. Chen, H., F. Wang, N. Helian and G. Akanmu, 2013. User-priority guided min-min scheduling algorithm for load balancing in cloud computing. Proceedings of the National Conference on Parallel Computing Technologies, February 21-23, 2013, Bangalore, India, pp: 1-8.
3. Aruna, M., D. Bhanu and R. Punithagowri, 2013. A survey on load balancing algorithms in cloud environment. Int. J. Comput. Applic., 82: 39-43.
4. Buyya, R., R. Ranjan and R.N. Calheiros, 2009. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. Proceedings of the 7th High Performance Computing and Simulation Conference, June 21-24, 2009, Leipzig, Germany, pp: 1-11.
5. Wickremasinghe, B., R.N. Calheiros and R. Buyya, 2010. Cloud analyst: A cloud sim-based visual modeller for analysing cloud computing environments and applications. Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, April 20-23, 2010, Perth, Australia, pp: 446-452.
6. Chen, W. and E. Deelman, 2012. Workflowsim: A toolkit for simulating scientific workflows in distributed environments. Proceedings of the IEEE 8th International Conference on eScience, October 8-12, 2012, Chicago, IL., USA., pp: 1-8.
7. Mahalle, H.S., P.R. Kaveri and V. Chavan, 2013. Load balancing on cloud data centres. Int. J. Adv. Res. Comp. Sci. Software Eng., 3: 1-4.
8. Domanal, S.G. and G.R.M. Reddy, 2013. Load balancing in cloud computingusing modified throttled algorithm. Proceedings of the IEEE International Conference on Cloud Computing in Emerging Markets, October 16-18, 2013, Bangalore, India.
9. Wang, S.C., K.Q. Yan, W.P. Liao and S.S. Wang, 2010. Towards a load balancing in a three-level cloud computing network. Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, July 9-11, 2010, Chengdu, China, pp: 108-113.
10. Randles, M., D. Lamb and A. Taleb-Bendiab, 2010. A comparative study into distributed load balancing algorithms for cloud computing. Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, April 20-23, 2010, Perth, Australia, pp: 551-556.
11. Kansal, N.J. and I. Chana, 2012. Cloud load balancing techniques: A step towards green computing. Int. J. Comput. Sci. Issues, 9: 238-246.
12. Ghafari, S.M., M. Fazeli, A. Patooghy and L. Rikhtechi, 2013. Bee-MMT: A load balancing method for power consumption management in cloud computing. Proceedings of the 6th International Conference on Contemporary Computing, August 8-10, 2013, Noida, India, pp: 76-80.