http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

OPEN ACCESS Information Technology Journal

ISSN 1812-5638
DOI: 10.3923/itj.2016.39.45

Research Article
A Priority Base Task Scheduling on Virtual Machines using Workflowsim

[1]Pradeep Singh Rawat, [2]Priti Dimri and [3]Varun Barthwal

[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]Department of Information Technology, Faculty of Information Technology, Hemwanti Nandan Bhuguna Garhwal University, Srinagar, India

Abstract
Computing Technology is changing the working culture in the differing organization. There is a paradigm shift by using technology as a utility by end users. Cloud computing paradigm is following the characteristics of utility base computing. Under the category of distributed computing, utility computing includes grid and cloud computing. This study highlights the efficient task scheduling at the virtual machine level. When tasks are scheduled over the virtual machine users need to allocate the tasks to the respective virtual machines. The virtual machine should have a good MIPS rating and better combination of other parameters e.g., ratio at the virtual machine level. For the high quality of service in cloud, computing virtualization plays an important role. The quality of service measurement depends on evaluation parameters used. In this study tasks, the finish time is used as evaluation metrics at scheduler level. Simulation technique is used for performance evaluation. Service model provided by the cloud service providers is modeled and simulated using cloudsim. Cloud service model, IaaS, PaaS, SaaS are simulated and user requests are handled by the datacenter across the globe. Simulation results are helpful for real deployment of cloud tasks. Simulated cloud environment is helpful for quality of service improvement. Cloudlet finish time is considered as a Qos parameter. Simulation results are improved with three cloud configuration scenarios with different MIPS rating.

Key words: MIPS, IaaS, PaaS, PaaS, VM, MINMIN, SLA, LBIMM, OLB

Received: February 25, 2016 Accepted: March 04, 2016 Published: March 15, 2016

Citation: Pradeep Singh Rawat, Priti Dimri and Varun Barthwal, 2016. A priority base task scheduling on virtual machines using workflowsim. Inform. Technol. J., 15: 39-45.

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 all types of users of cloud environment to retrieve the files and necessary documents. Files stored at remote location need to be accessed according to the service level agreement between the user and cloud service providers. The main issue is to schedule the incoming requests in an efficient way with a minimum finish time of tasks corresponding to the user requests. Different load balancing algorithms like FCFS, round robin, active-VM monitoring and throttled are used for executing client's request with a minimum response time and requests are assigned to the available virtual machines with light weight state (Al Nuaimi *et al*., 2012). It is essential at the service provider's end that every virtual machine in the cloud system should do the same amount of work throughout, i.e., minimizing the response time.

A load of a machine can be balanced by dynamically shifting the workload local to the machine to remote nodes or machines which are underutilize. Optimal configuration of virtual machine runs the task in minimum time span. The load balancing process maximizes the resource utilization and user satisfaction, minimizing response time. By reducing the number of job rejections and increasing the performance ratio of the system management of the dynamic resources in cloud platform can be efficiently given by virtualization technology. The power efficiency of the improvement of datacenter enables the assignment of multiple virtual machines (VMs) to a single physical server (Xiao *et al*., 2013). Consequently, some of the servers can be turned off or put into a sleep state, thereby, lowering the power consumption of the cloud computing system. In this study, a novel simulation configuration is presented for VM-assignment to the cloudlet. It can be helpful to allocate the incoming jobs to available virtual machines. This study provides the prerequisite for real deployment tasks on the real cloud. Here, the virtual machine assigned depends on its status i.e., overloaded or under loaded. If VM is found with the least loaded, then new request is allotted. This study highlights the simulation configured to create one scheduler, one data center and 20 virtual machines. For resource optimization and scheduling the tasks and virtual machine MINMIN local scheduling algorithm is used which uses MINMIN scheduling algorithm extends the properties of data aware scheduling algorithm.

MATERIALS AND METHODS

System model: This study is based on modeling and simulation techniques for cloud environment. Cloud environment is modeled by using programming abstraction with object oriented approach. Physical computing storage and computing nodes are the part of data center. Cloud main resource is modeled as an IaaS model, platform to run the cloud tasks PaaS model is simulated using cloudsim and cloud base application or user requests are handled as SaaS service model. Two basic approaches are focused for research which includes quantitative approach and the qualitative approach. The former approach is involved to generate the data in a quantitative form which may be applicable for rigorous quantitative analysis in a formal and rigid fashion. Key components of cloud environment are simulated for testing the new load balancing and scheduling policy. The simulation base approach may helpful for cloud service providers as well as cloud end users. The research study is based on simulation of cloud tasks using simulation tool. The experiment is carried out in the simulator. For the experimentation cloudsim based workflowsim simulator has been used (Kliazovich *et al*., 2013) (ref of workflowsim). As cloud infrastructure is distributed in nature requests will be coming from all geographical locations and should be handled intelligently. Workflowsim simulator gives the scenario with virtual machine configuration and host configuration at datacenter level. The simulator is flexible to provide virtual machines, data centers, bandwidth and much more for experimental parameters (Chen and Deelman, 2012). For experimental work parameters at the virtual machine level are shown in Table 1. Table 2 shows the host configuration and Table 3 shows the characteristics of the datacentre. End users requests are simulated in cloudsim following the SaaS model. The IaaS modeled which include the following entities i.e., data center, host with different processing elements. As shown in the configuration at platform level is simulated as PaaS model. For different user requests tasks are modeled using SaaS service model. Simulated task is executed on PaaS model which include the properties of virtual machine.

Table 1 describes the properties of a virtual machine which remain constant for different scenarios. All parameters value depends on available resources at host level within the datacentre. Tasks or cloudlets are scheduled on a virtual machine using an efficient scheduling algorithm. Simulation work simply uses the MINMIN algorithm for task scheduling on a virtual machine with a unique id. As per the requirement of an end user number of virtual machines used may vary from

Table 1: VM parameters

image size (MB)	1000	Task scheduling algorithm
VM memory (MB)	128	MIN-MIN
MIPS	10	
Bw	100	
No. of cpus	2	
VMM name	Xen	
No. of VMs	25	

Table 2: Host configuration parameters

Host id	Host memory (MB)	Host storage	Bw (MBps)	MIPS
0	2048	100000	10000	2000

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
Cost 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 band width within a data center in MBs)	15 MBs

1-25. "Xen" virtualization software is provided Microsoft for hosting virtualization. The virtual machine plays an important role in resource optimization.

Table 2 describes the host configuration at data centers. The host entity at the data center is virtualized to allocate the storage, computing and network resources to the cloudlets. All parameter's value used in Table 2 sets the limit of primary, secondary storage as well as a band width requirement at the user level.

Table 3 describe the datacenter characteristics which includes the software used for virtualization, pricing model, band width within the datacenter. In this simulation, datacenter characteristics are used with Xen for hosted virtualization architecture. Transmission rate within the datacenter indicates the communication rate for file transfer between different storage and computing server. The standard pricing model pay as you go, subscription models are used for Service Level Agreement (SLA) between cloud service providers and cloud end users.

RESULTS

Key findings of simulation results is based on finish time of cloud task deployed on PaaS and IaaS model. Parameters at PaaS level is fixed for 9 different scenarios. Quality of service parameter is improved along with the increasing value of ratio. For real deployment of applications simulated scenario for dependent task is helpful for service provider to fulfill the requirement of cloud user.

Table 4: Simulation results in 1

Ratio	Start time	Finish time	VM MIPS rating (VM parameter)
1.0	0.1	11.00	10
1.5	0.1	7.33	10
2.0	0.1	5.60	10
2.5	0.1	4.50	10
3.0	0.1	3.77	10
3.5	0.1	3.24	10
4.0	0.1	2.85	10
4.5	0.1	2.54	10
5.0	0.1	2.30	10

Table 5: Simulation results in 2

Ratio	Start time	Finish time	VM MIPS rating (VM parameter)
1.0	0.1	1.20	100
1.5	0.1	0.83	100
2.0	0.1	0.65	100
2.5	0.1	0.54	100
3.0	0.1	0.47	100
3.5	0.1	0.41	100
4.0	0.1	0.38	100
4.5	0.1	0.34	100
5.0	0.1	0.32	100

Table 6: Simulation results in 3

Ratio	Start time	Finish time (msec)	VM MIPS rating (VM parameter)
1.0	0.1	0.21	1000
1.5	0.1	0.17	1000
2.0	0.1	0.16	1000
2.5	0.1	N/A	1000
3.0	0.1	N/A	1000
3.5	0.1	N/A	1000
4.0	0.1	N/A	1000
4.5	0.1	N/A	1000
5.0	0.1	N/A	1000

Table 4 shows the simulation results for configuration parameters shown in the previous section. The quality of service parameter, i.e., finish time of cloudlet depends on 2 parameters, (1) Ratio in virtual machine level and (2) MIPS rating at the virtual machine level. When ratio varies from 1.0-5.0 then evaluation measurement parameter improved from 11.0-2.30 msec. The quality of service is improved with 79%.

Table 5 represents the simulation results for virtual machine parameters with constant MIPS = 100 and ratio varies from 1.0-5.0. The quality of service at the virtual machine level is improved. Cloud resources are fully utilized with QoS improvement is 73%.

Table 6 shows the better quality of service i.e., finish time = 0.16 msec but the ratio can vary between 1.0-2.0 with MIPS rating = 1000. For ratio greater than 2.0 it cannot be predicted by the quality of service parameters i.e., indicated by N/A in Table 6. From Table 4-6 it is clear that best pair for the quality of service parameters i.e., ratio = 2.0 and MIPS rating = 1000.

Table 7: Simulation scenario

Simulation configuration scenario	MIPS rating	Ratio	Finish time (msec)
1	10	5.0	2.30
2	100	5.0	0.32
3	1000	2.0	0.16

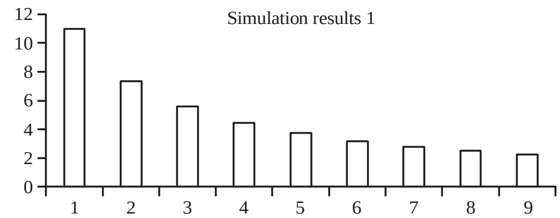

Fig. 1: Simulation configuration 1 with VM MIPS = 10, bars showing finish time

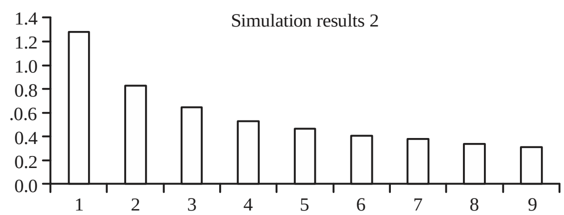

Fig. 2: Simulation configuration 1 with VM MIPS = 100, bars showing finish time

Table 7 represent the simulation result comparison for 3 different scenarios. The scenario with MIPS rating 1000 and ratio at virtual machine level = 2.0 provide the minimum finish time of cloudlet. For resource optimization, selection of the parameters is needed for real deployment in a cloud environment. The second and third scenario provides the best quality of service at the virtual machine level. Comparison of 3 scenarios is shown with graphical representation.

Simulation configuration comparison: Figure 1 presents the variation of cloudlet finish time with different values of the virtual machine ratio and constant virtual machine MIPS rating = 10. For VM ratio 5.0 minimum finish time = 2.30 msec has been found.

Figure 2 presents the variation of cloudlet finish time with different values of virtual machine ratio and constant virtual machine MIPS rating = 100. For VM ratio 5.0 minimum finish time = 0.32 msec has been found.

Figure 3 presents the variation of cloudlet finish time with different values of virtual machine ratio and constant virtual machine MIPS rating = 1000. For VM ratio 2.0, minimum finish time = 0.16 msec.

Figure 4 presents the comparison of cloudlet finish time with different values of virtual machine ratio and virtual

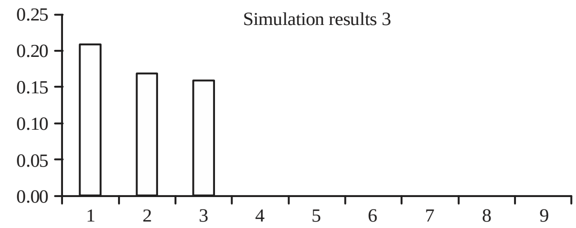

Fig. 3: Simulation configuration 1 with VM MIPS = 1000, bars showing finish time

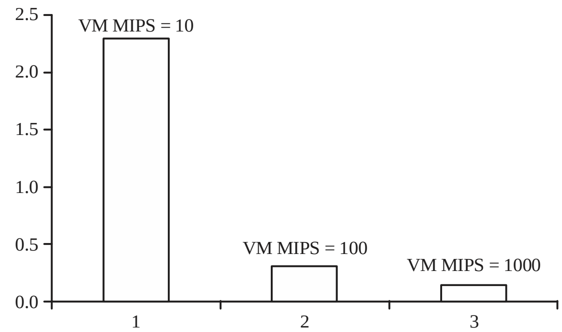

Fig. 4: Simulation configuration scenario comparison, bars showing finish time (msec)

machine MIPS ratings for 3 different scenario = 10, 100, 1000. For VM ratio 2.0 and MIPS rating minimum finish time = 0.16 msec has been found as shown in Fig. 4. Using these scenarios it can find that out the best combination of parameters for real deployment of cloudlets or cloud tasks.

DISCUSSION

The cloud facility is a network of geographically distributed datacenters across the globe. Each datacenter consists 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. Min scheduling algorithm (Chen *et al*., 2013). 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 the 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) (Chen *et al*., 2013). 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 produces it and the ready time of both resources are updated. The process is repeated 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 produce a schedule which improves load balancing and also reduces the overall completion time. But still it does not consider the priority of a job while scheduling.

User-priority aware load balance improved Min-Min scheduling algorithm (PA-LBIMM) (Chen *et al*., 2013). User priority is incorporated with LBIMM algorithm to develop PA-LBIMM. This algorithm will first divide all the tasks into two groups G1 and 2. The G1 is for the VIP user's tasks having higher priority requirement. G2 is for the ordinary user's 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 a 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. 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 a 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 (Mohialdeen, 2013). The data center controller (Wickremasinghe *et al*., 2010), 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. This study makes use of MINMIN task scheduling for resource provisioning at the virtual machine level. 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 (makespan) and will cause some bottlenecks since requests might be pending waiting for nodes to be free (Mahalle *et al*., 2013).

Mahalle *et al*. (2013) 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 data center controller sends the request to the VM identified by that id. Data center controller notifies the active VM load balancer of the new allocation. Domanal and Reddy (2013) 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 to the request and id of VM is returned to data center, 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 data center queries load balancer for allocation of VM. Wickremasinghe *et al*. (2010) have developed Throttled algorithm which is completely based on the virtual machine. Here the client first requests the load balancer to check the right virtual machine which accesses that load easily and perform the operations which are given by the client (Wickremasinghe *et al*., 2010). In this algorithm the client first requests the load balancer to find a suitable virtual machine to perform the required operation such a study consider 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) (Wang *et al*., 2010) 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. Two phase load balancing algorithm (Wang *et al*., 2010) 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 the manager. In the first phase OLB scheduling manager is used to assign the 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 a static environment. Honey bee foraging algorithm (Randles *et al*., 2010). It is a decentralized honeybee-based nature-inspired load balancing technique for self-organization. It achieves global load balancing through local server action (Kansal and Chana, 2012). 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 (Ghafari *et al*., 2013). Harvester bees then follow the foragers to the location of food to harvest it. In this approach, the servers are grouped under Virtual Servers (VS) each having its own virtual service queues. Each server processing a request from its queue calculates a profit, which is analogous to the quality that the bees show in their waggle dance. One measure of this reward can be the CPU time spends on processing a request. The dance floor in case of honey bees is analogous to an advert board which is used to advertise the profit of the entire colony. Each of the servers takes the role of either a forager or a harvester. The server after processing a request can post their profit with a probability pr. A server can randomly choose a VS's queue with a probability px. A server serving a request, calculates its profit and compare it with the colony profit and then sets its px. If this profit was high, then the server stays at the current virtual server. If it was low, then the server returns to the idle/waiting behavior. This approach works well with heterogeneous types of resources but it does not show equivalent improvement in throughput while increasing a number of resources. In order to ensure that a small peak instantaneous load does not trigger unnecessary migration load forecasting program is quite helpful. When an indicator of the node exceeds the threshold, instead of immediately triggering the migration, it predicts its future n load value according to its historical load recorded. When at least k values in the prediction are bigger than the threshold, the migration begins.

This study briefly summarizes the load balancing algorithms used in the cloud computing environment. The main focus is on the efficient utilization of the virtual machines and balancing the virtual machines with the incoming request. Datacenter model plays an important role for simulation of IaaS service model. Identification of best load balancing is defined as a process of making effective resource utilization. Novel approach is used for minimizing under or over utilization of the available resources or virtual machines. Mahalle *et al*. (2013) 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 data center controller sends the request to the VM identified by that id. Data center controller notifies the active VM load balancer of the new allocation. Domanal and Reddy (2013) 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 to the request and id of VM is returned to data center, 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.

Wickremasinghe *et al*. (2010) have developed Throttled algorithm which is completely based on the virtual machine. Here the client first requests the load balancer to check the right virtual machine which accesses that load easily and performs the operations which are given by the client (Wickremasinghe *et al*., 2010). 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 active-VM load balancer and proposed VM-assign algorithm for comparison is focused. 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. Priority of cloud resource is based on simulation results obtained by using best configuration and condor VM with different MIPS share of the virtual machine. Host at data center takes the responsibility to allocate the maps power to the VM associated with the virtual machine. With constant MIPS rating and variable ratio parameter at virtual machine level quality of service can be improved.

CONCLUSION

In this study, an efficient algorithm is used which manages the load at the server by considering the current status of the all available VMs for assigning the incoming requests intelligently i.e., MINMIN scheduling algorithm. Using this efficient algorithm it is tried to find out the optimal configuration at the virtual machine level and datacenter level. Finish time of cloud task is considered as a quality of service evaluation parameter. The simulation results are obtained by using cloudsim based cloud simulator i.e., workflowsim. The QoS is improved with good MIPS rating at VM level as well as ratio parameter is also taken into consideration. The value of ratio at VM level used i.e., 1.0-5.0 and MIPS rating from 10, 100, 1000 at the virtual machine level. It is found that for MIPS rating 100 and ratio 5.0 found 86.08% improvement in evaluation parameter. In the case of MIPS rating 1000 at virtual machine level ratio cannot exceed greater than 2.0 but ratio = 2.0 and MIPS = 1000 provides an optimal solution for real-time deployment of tasks i.e., finish time improved with 93.04%.

ACKNOWLEDGMENTS

We wish to thank Weiwei Chen; Deelman 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

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.

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.

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.

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 (CCEM), October 16-18, 2013, Bangalore, India..

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.

Kansal, N.J. and I. Chana, 2012. Cloud load balancing techniques: A step towards green computing. Int. J. Comput. Sci. Issues, 9: 238-246.

Kliazovich, D., S.T. Arzo, F. Granelli, P. Bouvry and S.U. Khan, 2013. e-STAB: Energy-efficient scheduling for cloud computing applications with traffic load balancing. Proceedings of the Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, August 20-23, 2013, Beihang University, Beijing, China, pp: 7-13.

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.

Mohialdeen, I.A., 2013. Comparative study of scheduling al-gorithms in cloud computing environment. J. Comput. Sci., 9: 252-263.

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.

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.

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.

Xiao, Z., W. Song and Q. Chen, 2013. Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans. Parallel Distrib. Syst., 24: 1107-1117.