ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

Refined VM-Assign Load Balancing in Cloud Computing

Prithpal Mohini Singh[1], Shaveta Angurala[2]

Computer Science Department, DAV Institute of Engineering and Technology, Jalandhar, India[1, 2]

Abstract: One of the major challenges in Cloud is the scheduling of the requests to get optimized (faster) response time and better performance in the Cloud Configuration. Various Load Balancing methods are developed to improve the efficiency of the Cloud. The paper aims to provide a combination of three different techniques as Refined VM-Load Balancer Algorithm; Batch Processing, Defined Value and Priority Algorithm, to balance the load and for proper utilization of resources in Cloud Computing environment. The paper also compares the Proposed Refined Load Balancer with the existing Optimal Load Balancer Algorithm to analyse the results using Cloud-Analyst simulator. The results showed improvements on response time improving the performance and efficiency of the Cloud.

Keywords: Cloud Computing, Scheduling, Load Balancing, Virtual Machines, Response Time, Utilization, Virtualization, Refined Load-Balancer, Batch, Priority.

I. INTRODUCTION

Cloud Computing is a distributed computing paradigm that focuses on providing a wide range of users with distributed access to scalable, virtualized hardware or software infrastructure over the internet. [3] Load balancing is considered as one of the challenges in cloud computing, it is the major factor to improve the performance of the cloud computing. The current load balancing algorithms in cloud computing environment is not highly efficient [9]. Load balancing in cloud computing environment is very complex task till today, because prediction of user request arrivals on the server is not possible. Each virtual machine has different specification, so it becomes a very difficult to schedule job and balance the load among nodes [10]. Hence, Various Load Balancing solutions were implemented to optimize the response and processing time for improving the performance of the Cloud.

Types of Cloud Computing are:

1) As Cloud Service Models:
Three types of Cloud Service Models are:
- Infrastructure as a Service (IaaS),
- Platform as a Service (PaaS), and
- Software as a Service (SaaS).

2) As Cloud Deployment Model:

a. Public clouds: Public clouds are termed as those in which service provider's offer their resources as services to the general public.
b. Private clouds: Also known as internal clouds, private clouds are designed for exclusive use by a single organization. A private cloud offers the highest degree of control over performance, reliability and security. [11] [12].
c. Hybrid clouds: A hybrid cloud is a combination of public and private cloud models that tries to address the limitations of each approach. In a hybrid cloud, part of the service infrastructure runs in private clouds while the remaining part runs in public clouds [11] [12].

One of the challenges of Cloud Computing is the scheduling of incoming User/ Client requests so as to improve the efficiency and effective utilization of resources. Scheduling, allocation and processing of request to optimize the response time the scheduler takes to process the task and de-allocate to acquire other tasks, Load balancing is introduced. The aim of load balancing is to devise an algorithm which properly utilizes its resources and handles the user/client requests effectively, and selects the best resource or VM to serve the task or request, hence reduces the under or over-utilization of resources to help servers or data centre.

The load is distributed to maximize the efficiency of the algorithm. Earlier Scheduling algorithms were unable to prioritize requests. Load balancing is defined as the process of reassigning the available load to the individual nodes in the specific data centers of the collective cloud environment to improve the resource utilization and job response time of the system. It also sorts out the situation where some of the nodes in the cloud are heavily loaded while other nodes are idle or under-utilized. Load balancing ensures that all nodes in the system approximately equal amount of work at any instance of time [4, 9]. The Objective of Load Balancing is to achieve optimum resource utilization, maximize throughput, minimum response time.

In this paper, the Refined VM-Assign Load Balancing Algorithm describes the implementation to balance the load on the cloud for optimized results which lowers the response time, improves performance and utilizes the resources available in best possible way to improve the allocation and servicing of user-based requests.

The Refined Algorithm defines batch method first to categorize the number of user-requests, to specify maximum load capacity of individual VM and the batch size to be allocated calculates and compares Defined values and prioritizes VM for allocation.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

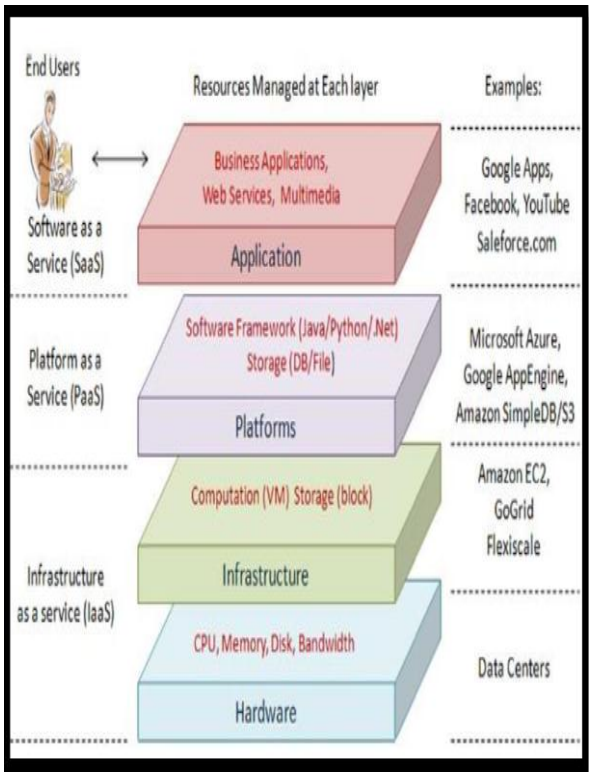

Figure 1 Types of Cloud Computing: Cloud Service Model.

Section II briefly describes the types of algorithm that have been implemented to balance load of incoming requests to optimize response time and performance in Cloud Computing environment. Section III describes the proposed technique of Refined-VM Load Balancing Algorithm, which implements Batch function for determining the number of user requests, the maximum allocation capacity of individual VMs to serve user requests, and the batch size of user-requests to be allocated to the available VM. The Current Allocation values on every VM are calculated and comparisons are made with maximum allocation capacity to get defined value of resources. Then, it prioritizes the VMs according to the comparisons made for allocating the batch of user requests/ jobs to specific or selected VM/Resource. The remaining sections provide the details to implement the Proposed Algorithm and the analysis and comparison of the loading balancing techniques.

II. BACKGROUND AND RELATED WORK

The Load Balancing Algorithms are defined to minimize the response time and for effective utilization of resources or virtual machines, which results in better performance and output and faster processing of requests and allocation of resources. Load Balancing algorithms are categorized as static and dynamic load balancing algorithm.

Static Load Balancing technique divides the load equally between the available servers at the compile time. The host to serve the request is selected during the process creation and cannot be change at run-time of the process. Round Robin is an example of Static Load Balancing method. [7]

Dynamic Load Balancing deals with the requests of client at run-time or at the execution time the host/resource selection takes place depending on the conditions like underutilized resource or the least loaded or least recently used host or VM to serve and allocate the user-requests or job. It is based on the current state of the system. Two types of Dynamic Load Balancing are: [6] [7]

1) Distributed dynamic load balancing:
In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. A benefit, of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt; it instead would affect the system performance to some extent.

2) Non-distributed dynamic load balancing:
In the non-distributed there is one node responsible for load balancing of the whole system. The other nodes interact merely with the central node [8]. The Chronological presentation of the related works and existing Load Balancing techniques are as follows:

Hu et al in [15] proposes algorithm for enhancing job scheduling using genetic information by mapping historical data and current state of the system. It chooses the least-affective solution by computing ahead influence of the system after the deployment of the needed VM resources, finds the best scheduling solution using population. The experimentation results show an improvement in the utilization of resources. On the other hand, it has high cost to store and retrieve the historical data, and this may also increase the response time.

Fang et al in [14] obtains high resource utilization and meet dynamic requirements of task by providing a two level task scheduling. The paper improves the response time, resources utilization by mapping task to VMs and then VMs to host resources. This approach may have succeeded in improving the resource utilization, but that using two levels of task scheduling would increase the response time compared with other load balancing algorithms.

Sethi et al in [16] introduce a load balancing algorithm using fuzzy logic with Round Robin (RR) algorithm. The algorithm is based on various parameters such as processor speed, and assigned load in VM and etc. The algorithm maintains the information of each VM and numbers of requests currently allocated to VM. For one VM least loaded VM is searched. Processor speed and load in VM using fuzzy logic for more than one VM hence enhancing the performance and decreasing the response time, performance is better than RR algorithm. The drawback of this algorithm is they didn't consider about processing cost, and compared their results with only RR algorithm which had been enhanced and improved by many researchers before.

Subramanian et al in [17] propose a new algorithm that combines the advantages of three algorithms: greedy, round robin, and power saver algorithm, and overcomes their disadvantages. It focuses on best utilization of resources and minimizes the power consumption. It

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

scheduled the VMs to the nodes depending on their priority value, which varies dynamically based on their load factor. When a request is received, the node with the maximum available resource is determined and then it is checked whether the node had a load factor less than 80%. If the highest priority node had a load factor less than 80%, then the VM is scheduled to that node, otherwise it checks the next maximum resource.

Shridhar G. Damanal in [1] VM-assign load balancer algorithm maintains an index assign table of virtual machines and also the load of VMs. They have made an attempt to efficiently use the available virtual machines depending on its load by selecting a VM for processing client's request. It checks for least loaded VM. Initially all VM are free so it follows Round Robin. Then if next request comes then it checks for VM table, if the VM is available and it is not used in the previous assignment then, it is assigned with the request and id of VM is returned to Data Center, else we find the next least loaded VM and it continues and follows the above step. In this way, a number of comparisons and condition checks are made which consumes more time and memory, thereby increasing the complexity of algorithm to allocate a single user-based request/job to the list of available VM. The proposed algorithm aims to reduce such cumbersome comparisons and conditional assignment of VMs by introducing the combination of three techniques; Batch function, Defined Value and Priority of VM based on the Defined Value. The number of user-requests determines the batch size of the Algorithm.

Terms/ Variables	Definitions
Userreqn	Number/ count of user-requests/jobs for VM allocation from user/client
Nb	Batch-size of the user-request/jobs, depending on the number of user-requests to be served (5% of userreqn).
Maxalloccap	The maximum value of user-requests that can be allocated to of individual VMs (90% of user-requests).
VMs	Number of VMs in the cloud.
Currallocval	The value of current user-requests that every individual VM is allocated.
Max(currallocval)	This function dynamically selects the maximum value out of the currently allocated user-requests/jobs on VMs.
Priority	Assigns integer value from 1 to the number of VMs in the cloud. It is initialized to 1.

Table 1.Terms and their definitions

In the present work we are comparing Optimal Load Balancer and Propose VM-assign Load Balancer Algorithm to ensure that resources aren't under-utilized or over-utilized by reducing the number of comparison and condition checks thereby optimizing the response time and processing time to improve the performance of the Cloud.

Refined VM- Assign Load Balancer Algorithm

Input: No of incoming user-requests R1, R2 . . . Rn.
Available The number of VMs; VM1, VM2 . . . VMn.
Output: All incoming user-requests R1, R2 . . Rn are allocated according to the batch function and priority is assigned according to DefVal of individual resources among the available VMs; VM1, VM2 . . . . . VMn.

1: Initially all the VM's have 0 allocations.

2: Refined VM-assign load balancer maintains the Allocation table of VMs which has no. of requests currently allocated to each VM, DefVal, Priority of VMs.

3: When user requests arrive at the data-centre, it passes the jobs over to the load balancer.

4. The function Batch is defined, Batch (UserReqN, NB, MaxAllocCap)
(
If (UserReqN <= 100) then NB = 5 AND MaxAllocCap = 90.
If (UserReqN <= 1000) then NB = 50 AND MaxAllocCap = 900.
If (UserReqN <=10000) then NB = 500 AND MaxAllocCap = 9000.
..... …
If (UserReqN<=n) then NB = .05n AND MaxAllocCap = .9n;
)

5. Calculate the CurrAllocVal on every VM.

6. For every VM/Resource, DefValue = MaxAllocCap – CurrAllocVal.
a. For Every VM, Compare the DefValue with the Max(CurrAllocVal)
Check and parse table For (Max(DefVal) = priority; priority++);
Select Max(DefVal); assign VM(priority).
Return to Step a.
b. Allocate the Batch() to the VM(highest_priority).
Update VM (CurrAllocVal).
Return to Step 5.

7. Response is received at the Data-Center after VM has finished the user-request/job.

8. The data canter notifies the Refined VM-assign load balancer for the VM de allocation and updates the table; Return to Step 2.

DOI 10.17148/IJARCCE.2015.41019

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

III.PROPOSED REFINED VM-ASSIGN LOAD BALANCING ALGORITHM

The Proposed Algorithm Refined-VM Load Balancer Algorithm combines batch processing, Priority Algorithm and Defined Value to get optimized response time, reduced cost and to maximize performance. It overcomes the flaw of not assigning the next/ incoming user-request to the VM which was allocated in the previous assignment, as batch processing is used to allocate requests which saves time of comparison of individual VMs apart from checking for their availability, current allocation time and whether it was used in the previous request or not. In this case the response time of the algorithm could degrade and hence reduce the performance of the Cloud. The Refined Load Balancer firstly checks for the number of user-requests to be served and allocated though a function called Batch, which specifies the batch size and the maximum capacity of load allocation on individual VMs/Resources based on number of jobs/user-requests received.

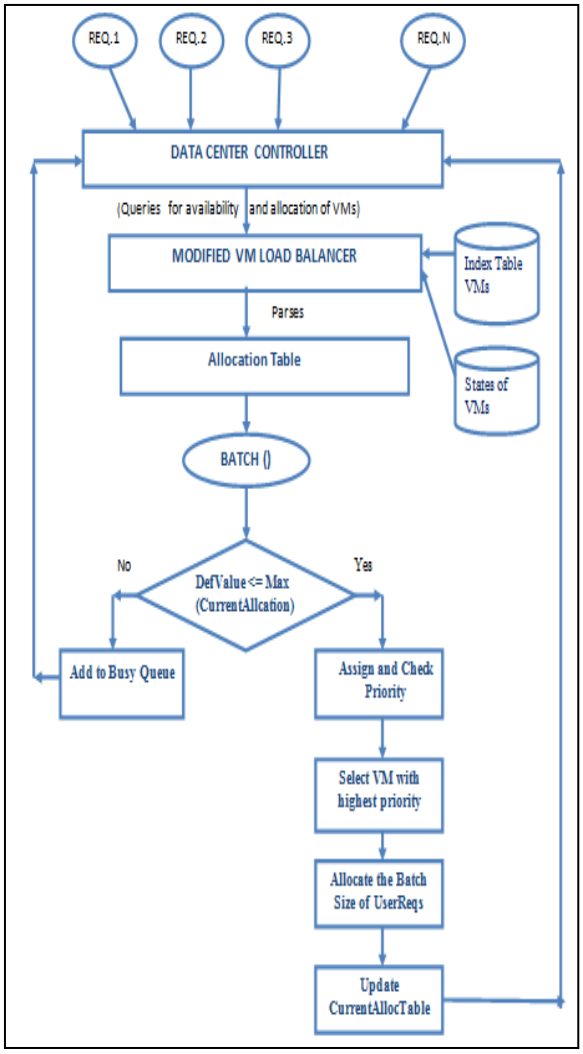

Figure2. The flowchart refined load balancing algorithm.

The Algorithm then calculates the current allocation of individual VMs and selects the maximum value of current allocation count. The defined value is calculated for every VM by deducting the Current allocation size from the maximum capacity that can be allocated to every VM (maximum capacity of individual VMs is decided by the batch function depending on the number of user-requests received). The defined value is then compared with the Maximum current allocation value for every VM, to set priority on individual VMs. The highest defined value means lesser user-requests are allocated to the VM, hence higher priority.

So the higher defined values get high priorities to which batch of user-requests/jobs can be allocated. After allocating the VM, it updates the allocation table and calculates the defined values again for comparison and prioritizes the VMs for task/job allocation. Instead of parsing the table, checking the availability status, and the last assigned VM every for single user-request as specified in Optimal Scheduling. The proposed Algorithm parses Allocation table for batch of user-requests, thereby reducing the number of comparisons and certain conditions which leads to faster response time and better performance of the system.

The Flowchart in Fig.3 briefly describes the Flow of the Proposed Algorithm – Refined VM-Assign Load Balancing and the three main techniques that are implemented in the algorithm to use available resources efficiently, optimize the response time, reduce overheads, and improve the overall performance of the Cloud environment.

IV. EXPERIMENTAL SETUP

The Proposed Algorithm is simulated and implemented by using Eclipse IDE and Cloud Analyst in Windows7 (/Linux) platform to get the desired results. The Case Studies are developed to where the number of VMs and user-requests are increased over time to display the scalability factor for the proposed algorithm. The number of VMs is increased, in this Algorithm we consider four cases to analyse the performance of the algorithm for optimized results.

Cloud Analyst is a graphical simulation tool based on Cloudsim for modeling and analyzing behavior of cloud computing environments, which supports visual modeling and simulation of large-scale applications that are deployed on cloud Infrastructures [18].

The Cloud Analyst allows setting location of users, number of user and number of request per use per hour. And also it allows setting the location of the data centers, number of virtual machines, number of processors, amount of storage, network bandwidth and other necessary parameters [5]. The world is divided into 6 regions based on the 6 main continents in the world. The other main entities such as user bases and data centers belong to one of these regions [19]. A user base's main responsibility is to generate traffic for the simulation. A single user base may represent thousands of users but is configured as a single unit and the traffic generated in simultaneous bursts representative of the size of the user base. Datacenter is used to control the various data center activities [20] such as VM creation and destruction and does the routing of

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

user requests received from user bases via the Internet to the VMs [19]. We should minimize the response time in order to enhance the system performance. The response time can be obtained by:

Response time = the users request processing delay + Network delay (1)

Four Case Studies are considered by scaling the number of VMs from 5 to 75 and increasing the number of User-request/jobs to compare the proposed algorithm with the Optimal Load Balancing Technique. Each Data Center has a capacity to host a no of virtual Machines (from 5 to 75) which are needed for particular application. The application deployment Configuration specifies Data Center (DC1- DC5 at different geographical locations), number of VMs, the image size10k, memory size 512, BW 1000 in Main Configuration Section. The Data Center Configuration can be used to add new Data Center or to remove an existing Data Center.

Case Study: No. of VMs	Optimal Load Balancer	Refined VM Load Balancer	%Optimization in response time (approx)
VM = 5	382.31	243.23	**36.38%**
VM = 25	1112.79	682.38	**38.66%**
VM = 50	388.78	261.19	**32.73%**
VM = 75	355.48	240.81	**32.39%**

Table2. Comparison: Average Response Time for Scheduling Algorithm and the percentage improvement of the response time.

V. RESULT ANALYSIS

The Results are analyzed with respect to the efficient utilization of resources and the response and processing time of proper utilization of the virtual machines/resources by avoiding the under or over loading conditions of the load on all available VMs in an efficient way. Hence we can say that our algorithm will overcome the under-utilization of resources.

In the first case study, the specifications for Application Deployment Configuration consist of five Data Centers at different geographical locations each having five Virtual Machines/ Resources, service broker policy as 'optimize response time'. The simulation time is sixty minutes and a specified number of User-bases are added in different regions. The Optimal Load Balancer completes the execution of tasks at an average response time of 382.31 ms, whereas for proposed Refined Load Balancer Algorithm the response time is 243.33ms. The response time for serving user-requests is reduced by approximately 32% in this case. When the Virtual Machines are increased to 25, the Optimal Load Balancer completes the execution of tasks at an average response time of 1112.79 ms, whereas for proposed Refined Load Balancer the response time is 682.38. The response time for serving user-requests is reduced by approximately 40% in this case. The number of Virtual Machines is increased to fifty each. The Optimal Load Balancer completes the execution of tasks at an average response time of 388.78ms, whereas for proposed Refined Load Balancer Algorithm the response time is 261.19ms. The response time for serving user-requests is reduced by approximately 32% in this case. When the of Virtual Machines to are increased to seventy-five each, the Optimal Load Balancer completes the execution of tasks at an average response time of 355.48ms, whereas for proposed Refined Load Balancer, the response time is 240.81ms. The response time for serving user-requests is reduced by approximately 32% in this case.

VI .CONCLUSION

In this paper, the Refined Load Balancer efficiently selects and prioritizes the allocation and serving of user-requests or jobs based on the current allocation and the maximum serving capacity of the Scheduling algorithm to balance load so as to utilize resources to its maximum capacity to reduce response time, switching and comparison time and improve performance. We have proved that the proposed algorithm is efficient in balancing of load by scheduling the execution of user-requests, providing another optimized technique to solve the problem of utilization of resources for faster processing of requests. Analysis the results of the various Case Studies for comparing the Proposed Technique with the existing Optimal Load Balance Algorithm, the results display the lower response and data processing time when run under similar Cloud Configuration.

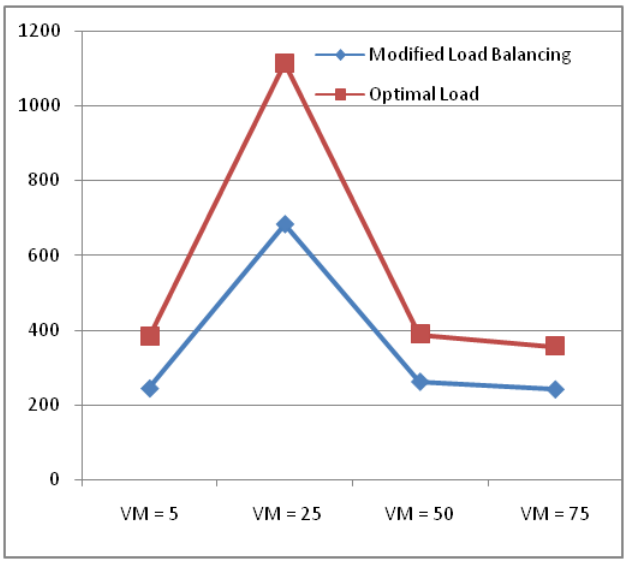

Figure 4 Graph: Comparison Response Time Refined and Optimal Load Balancing

From Comparison table, we can deduct that for similar Cloud Configuration Environment, the response time is optimized and decreases by 32-40% under various case studies. As future scope, more VMs can be added at different locations and remote user-requests can be served using nearest and least loaded VM. In such case, it will consider the propagation time/delay and response time required serving requests at a remote location or it can be

DOI 10.17148/IJARCCE.2015.41019

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

further improved by increasing the number of user-requests or jobs.

REFERENCES

[1] Shridhar. G. Donamal, "Optimal Load-Balancing in Cloud Computing by efficient utilization of virtual resources", IEEE 2014.
[2] Obaid Bin Hassan et al, "Optimal Load Balancing of Cloudlets using Honey Bee Behaviour Load Balancing Algorithm", International Journal of Advance Research in Computer Science and Management Studies, March 2015.
[3] Michael Armbrust, Armando Fox, Gunho Lee, Ion Stoica(2009) "Above the Clouds :A Berkeley View of Cloud Computing" University of California at Berkeley Technical Report No. UCB/EECS- 2009-28.
[4] Sharma, T. and Banga, V.K., Efficient and Enhanced Algorithm in Cloud Computing. International Journal of Soft Computing and Engineering (IJSCE),(March 2013)
[5] Mohapatra, S., Rekha, K.S., and Mohanty, S., A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing. International Journal of Computer Applications, 68(2013).
[6] Ratan, M. and Anant, J., Ant colony Optimization: A Solution of Load Balancing in Cloud. International Journal of Web & Semantic Technology (IJWesT), III,(2012).
[7] Deepika, Wadhwa, D., and Kumar, N., Performance Analysis of Load Balancing Algorithms in Distributed System. Advance in Electronic and Electric Engineering, 4(1): pp. 59-66,(2014).
[8] Mehta, R., Yask, P., and Harshal, T., Architecture for Distributing Load Dynamically in Cloud Using Server Performance Analysis Under Bursty Workloads. 1(9), (2012).
[9] Singh, A., Gupta, S., and Bedi, R., Comparative Analysis of Proposed Algorithm With Existing Load Balancing Scheduling Algorithms In Cloud Computing. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 3(1): pp. 197-200,(2014).
[10] Tiwari, M., Gautam, K., and Katare, K., Analysis of Public Cloud Load Balancing using Partitioning Method and Game Theory. International Journal of Advanced Research in Computer Science and Software Engineering, 4(2): pp. 807-812,(2014).
[11] Zhang, Q., Cheng, L., and Boutaba, R., Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1): pp. 7-18,(2010).
[12] Sajid, M. and Raza, Z. Cloud Computing: Issues & Challenges. in International Conference on Cloud. pp.35-41,(2013).
[13] [13] Hu, J., Gu, J., Sun, G., and Zhao, T. A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment. in 3rd International Symposium on Parallel Architectures, Algorithms and Programming. IEEE, pp. 89-96, (2010).
[14] [14] Fang, Y., Wang, F., and Ge, J., A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. Lecture Notes in Computer Science, Jg. 2010(6318): pp. 271-277,(2010).
[15] Sareen, P., Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of IT Governance in Cloud. International Journal of Advanced Research in Computer Science and Software Engineering, 3(3): pp. 533-538,(2013)
[16] Sethi, S., Anupama, S., and Jena, K., S, Efficient load Balancing in Cloud Computing using Fuzzy Logic. IOSR Journal of Engineering (IOSRJEN), 2(7): pp. PP 65-71,(2012).
[17] Subramanian S, N.K.G., Kiran Kumar M, Sreesh P, and G R Karpagam, An Adaptive Algorithm for Dynamic Priority Based Virtual Machine Scheduling in Cloud. International Journal of Computer Science Issues, 9(6),(2012).
[18] Pakize, S.R., Khademi, S.M., and Gandomi, A., Comparison of CloudSim, CloudAnalyst And CloudReports Simulator in Cloud Computing. International Journal of Computer Science And Network Solutions, 2: pp. 19-27,(2014).
[19] Mishra, R.K. and Bhukya, S.N., Service Broker Algorithm for Cloud-Analyst. International Journal of Computer Science and Information Technologies, 5 (3): pp. 3957-3962,(2014).