Proceedings of the SMART–2016, IEEE Conference ID: 39669
5th International Conference on System Modeling & Advancement in Research Trends, 25th–27th November, 2016
College of Computing Sciences & Information Technology, Teerthanker Mahaveer University, Moradabad, India

Enhanced Active Monitoring Load Balancing Algorithm for Virtual Machines in Cloud Computing

Shikha Garg[1], Dr. D.V. Gupta[2] and Dr. Rakesh Kumar Dwivedi[3]
[1]Research Scholar, Uttarakhand Technical University, Dehradun, India
[2]Professor, College of Engineering Roorkee, Roorkee, India
[3]Professor, Teerthanker Mahaveer University, Moradabad, India
E-mail: [1]shikha.incoer@gmail.com, [2]dvgupta_1@yahoo.co.in, [3]dwivedi.rakesh02@gmail.com

***Abstract*—This present research work aims to maximize the performance of cloud systems while reducing the response time through dynamic load balancing algorithm in cloud computing environment. Cloud computing provides different information technology services as a commodity to its users. When it was introduced, business industries were using large-scale mainframes and they are increasing in numbers and sizes. All are aims to increase its number of users to get more revenue. Hence they need different types of services having infrastructure, platform, software and many more in less response time. Two types of policies exist in cloud computing systems namely static load balancing and dynamic load balancing. Static load balancing policies do not consider the current state of system. These algorithms are suitable for homogeneous and stable environment. Dynamic load balancing depends on the present behavior of the system. There are various dynamic load balancing policies for virtual machines already existing in this context including Throttled and Active Monitoring in cloud computing systems. These types of algorithms are more flexible in nature. In this research paper, Enhanced Active Monitoring Load Balancing (EAMLB) algorithm is designed to minimize the response time in cloud systems. EAMLB policy provides better response time than Active Monitoring policy that improves the performance of cloud systems. Further, to assess the proposed algorithm, CloudAnalyst tool is used and comparative analysis is also represented.**

Keywords: Cloud Computing, CloudAnalyst Tool, Data Center, Load Balancing, Virtual Machine

I. INTRODUCTION

The term cloud computing is used in relation to Internet. The overall Internet can be viewed as a cloud. It is an important aspect to scale up the business without investing a high cost as it can fulfill almost all types of services including software, platform, infrastructure, storage on Internet at a very reasonable cost. Operational costs can also be reduced by using cloud computing [1].

Cloud computing has the concept of virtualization. With the help of virtualization end user can utilize different services in cloud. Data centers in cloud computing deliver services in the form of virtualization. A number of virtual machines (VMs) are created on data centers. There is need of efficient allocation of incoming request on these virtual machines so that their capacity can be better utilized [1,2,3]. There are also other parameters that should be better in terms of cloud systems. These parameters specially include performance, scalability, resource utilization, quality of service and response time.

Response time is one of the important factors in case of performance of cloud computing. Minimum response time increases the quality of service to the users and they can better utilize the services provided by the cloud. Enhanced Active Monitoring Load Balancing (EAMLB) algorithm is designed by keeping this parameter in mind. It enhance the concept of already existing Active Monitoring Load Balancing (AMLB) algorithm.

To assess the proposed algorithm CloudAnalyst tool is used. CloudAnalyst tool is GUI based tool and it is used to simulate large-scaled distributed applications under various deployment configurations on the Cloud. CloudAnalyst separate the simulation experimentation exercise from a programming exercise. A modeler can execute simulations repeatedly and can generate a sequence of simulation experiments with small changes in parameters [4,5,6].

In CloudAnalyst GUI refers the collection of various screens that enable the user to define parameters for simulation, characteristics of data centers, user base configuration and Internet characteristics. It also has the option to save and load the filled configurations and option to run the simulation. After simulation, it provides the result as graphical format. One of GUI screens is shown in Fig. 1.

 ISBN: 978-1-5090-3543-4

Fig. 1: CloudAnalyst GUI Screen

II. Related Work

This section describes the related work that has been done in the area of cloud computing to improve the performance of cloud systems using dynamic load balancing algorithms. The literature review of various proposed dynamic load balancing algorithm is briefly explained below.

In [2] load balancing algorithm is proposed that handles multi- dimensional and hierarchical resources in cloud computing systems. It constitutes the virtualization of storage and server in data centers. Application data and computation is tracked by this algorithm. It monitors the usage of resources including storage nodes, servers and switches on network. Author proposed the name VectorDot for this algorithm. In [7] researchers proposed a Stochastic hill climbing approach for the load balancing to maximize the optimization of resources on cloud computing. They have presented it as soft computing based load balancing approach. They showed the performance of algorithm with the help of simulation and found that this algorithm gives minimum response time than First Come First Serve (FCFS) and Round Robin (RR).

In [8] authors proposed a Weighted Signature based load balancing (WSLB) algorithm that minimizes response time to the user. They described using simulation on CloudAnalyst tool that WSLB outperforms the existing Round Robin, Equally Spread Current Execution (ESCE) load and Throttled algorithms in terms of response time. In [9] authors developed an algorithm namely Enhanced Equally Load Balancing Algorithm (EEDLBA) that dynamically allocates the load on virtual machines and they showed with the help of simulation that their algorithm gives better response time in comparison with other algorithm as Round Robin, server based load for internet distributed services, Scheduling Strategy on Load Balancing of Virtual Machine Resources, Central Load Balancing Policy for Virtual Machines and task scheduling algorithm.

In [10] researchers proposed a load balancing algorithm for heterogeneous cloud system to balance non preemptive independent task. They named it as Honey Bee Behavior Inspired Load Balancing (HBB-LB) that helps to balance load among virtual machines and maximize throughput. It also balances the priorities of tasks on the virtual machines so that the amount of waiting time for task can be minimal. They compared their proposed algorithm with existing load balancing and scheduling algorithms using simulation and showed that it gives better results. The researchers only use priority as the main quality of service QoS parameter. However, algorithm can be improved by considering other QoS factors.

Authors in [11] proposed dynamic load balancing algorithm which is based on the concept of least frequently used mechanism. They described that data center processing choice is made upon virtual machine. They compared the performance of algorithm with existing Round Robin and Throttled algorithm and found that it gives better response time. Authors suggest that in order to get better performance in cloud computing this algorithm can be combined with priority based and least recently used algorithm.

Authors in [12] represented a summary of swarm based and evolutionary algorithms including algorithm based on Ant Colony Optimization (ACO), Particle Swarm Optimization, Artificial Bee Colony (ABC) and Genetic Algorithm (GA) with their application and suitability to specific areas. They encourage the authors to develop some more algorithms to reduce time in cloud computing through load balancing. They studied various algorithms with their key objectives and issues as they studied Genetic Algorithm (GA) based techniques proposed in [13, 14] that are based on resource utilization and their key issues are to minimize the cost for communication, time for execution and ensures quality of service.

In [15] authors presented the performance analysis of load balancing algorithm in heterogeneous cloud computing environment. They have analyzed two algorithms Round Robin and Throttled in combination with three different service broker policies. They explained with the help of simulation using CloudAnalyst tool that Throttled algorithm with optimize response time gives best performance in heterogeneous cloud computing. They have not implemented theses algorithm in real time environment. In [16] researchers compared various static and dynamic algorithms present in cloud computing. They presented the comparison based on their challenges persent in cloud computing. They found that in many algorithms Load Balancing Min-Min (LBMM) uses less response time and it is more efficient in terms of resource parameter utilization.

 ISBN: 978-1-5090-3543-4

III. Existing Load Balancing Algorithm in CloudAnalyst Tool

To evaluate the performance of various algorithms, real environment is not easy to develop. So the simulation tool CloudAnalyst is used to study the existing and proposed algorithms. It provides the simulation in large scale cloud environment that can be set up according to different values of parameters. The existing load balancing policies in CloudAnalyst tool are as follows [5,6,8]:

A. Round Robin Load Balancing

Round Robin (RR) algorithm uses the round robin fashion to assign the incoming request to the virtual machines. It simply uses the time slice for each VM and other VM has to wait for its turn for allocation of request. In this, round robin scheduler has to manage the extra overhead of time slice for different VMs. It is the simplest load balancing policy defined in CloudAnalyst tool.

B. Active Monitoring Load Balancing

Active Monitoring Load Balancing (AMLB) algorithm maintains an index table that keeps the allocated VM id and total number or task or request allocated to that VM. When a new request arrives to the data center controller then AMLB search the least loaded VM but each and every time from the first index of table and assigned the task to that VM. In case if two VMs have same number of allocation than first VM is assigned.

C. Throttled Load Balancing

This is the third algorithm in CloudAnalyst tool. In this policy of load balancing, an index table having the id of VM along with its state either it is AVAILABLE or BUSY. When any task has to be allocated to the VM, then first it checks its state. If VM state is available then task is allocated and assigned VM id is returned to the data center controller that indicates the balancer to update the index table, if VM is busy then it task has to wait in queue.

IV. Proposed Work

This section contains the description of proposed work. An algorithm is developed by enhancing the concept of Active Monitoring load balancing (AMLB) algorithm, which finds the least loaded virtual machine among all virtual machine. In proposed Enhanced Active Monitoring Load Balancing (EAMLB) algorithm, Enhanced Active VmLoad Balancer also considers the virtual machine which is allocated recently along with the least loaded. It gives the benefit that one VM will not be allocated in continuous manner if it is least loaded. One VM needs not to process the task again and again. This minimizes the response time and increases the performance of cloud systems.

In proposed EAMLB load balancing policy, inputs are given as different jobs j1, j2,……., jn submitted by the user and the virtual machines v1,v2,….,vn created on data centers. In this, two hash maps are created. First is vmStateList that manage the total number of available and non allocated VMs and second is current Allocation Counts that holds the VMs with their allocation counts. When Data Center Controller (DCC) receives a new request then it queries to the load balancer for new allocation.

The load balancer first checks in vmStateList either any VM is available or not. If VM is available then allocate the job to it, otherwise check the least loaded VM which is not recently loaded. In this case load balancer moves to the next least loaded VM and allocate the request on it. The algorithm has been coded in Java and simulated in CloudAnalyst tool. Different variables are used in this algorithm as vmId that refers the Vm which has to be allocated, tempVmId that stores the recent allocated vmId, minCount that refers the minimum value of allocation that any VM can have, currCount that stores the current number of allocation for virtual machine which has to be checked for leas load.

Algorithm: EnhancedActiveVmLoadBalancer
Input: Jobs j_1, j_2,……,j_n to be assigned to virtual machine,
virtual machines v_1,v_2,….,v_n available at Data Center
Output: All submitted jobs j_1, j_2,……,j_n are allocated to available virtual machines

1: Create a Hash Map vmStateList<vmid, VirtualMachineState> and currentAllocationCounts <vmid, currentActiveAllocationCount>
2: DCC receives a new request
3: DCC queries the EnhancedActiveVmLoadBalancer for next job allocation
/* EnhancedActiveVmLoadBalancer starts checking for least loaded VM and also recently used VM */
4: vmId = -1
5: tempVmId = -1
6: IF currentAllocationCounts.size() < vmStatesList.size() THEN
7: FOR each VM in vmStatesList
8: IF currentAllocationCounts does not contain VM THEN
9: vmId = index value of checked VM
10: END IF
11: END FOR
12: ELSE
13: minCount = Integer.MAX_VALUE
14: FOR each VM in currentAllocationCounts
15: INITIALIZE currCount to the total number of current allocation of VM

16: IF currCount < minCount THEN
17: minCount = currCount
18: IF VM is not tempVmId THEN
19: vmId = index value of checked VM
20: END IF
21: END IF
22: END FOR
23: END IF
24: tempVmId = vmId
25: Allocate job to vmId
26: Return vmId to the DCC
27: DCC notifies EnhancedActiveVmLoadBalancer to update current allocation count table accordingly
28: IF VM finishes the job processing and DCC receives the response cloudlet THEN
29: DCC indicates EnhancedActiveVmLoadBalancer for VM deallocation
30: END IF
31: IF more request/job exist THEN
32: Repeat from Step 3
33: END IF

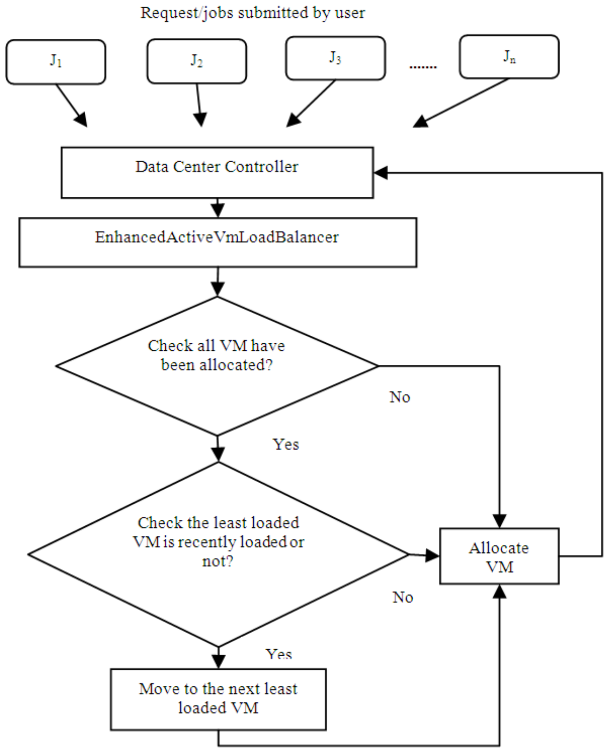

Fig. 2: Work Flow of EAMLB

In above algorithm, some abbreviations are used as tempVmId that temporary holds the recently allocated VM id, minCount that refer the minimum count of jobs that a VM can handle, currCount that have current number of jobs allocated to the VM, vmStateList that keeps the VM id with its state AVAILABLE/BUSY and currentAllocationCount that holds the VM id with its number of current allocation. vmStateList and currentAllocationCount are hash map.

The workflow of Enhanced Active Monitoring Load Balancing (EAMLB) algorithm is depicted in Fig. 2.

V. Result Analysis and Discussion

Enhanced Active Monitoring Load Balancing (EAMLB) algorithm is coded in Java language. After the coding of algorithm, the code is compiled by JDK version 6 and generated class file is used for implementation in CloudAnalyst tool. The main configuration parameters in CloudAnalyst tool are described in Table I and data center configuration is shown in Table II. Three data center are considered with same configuration and 5 VMs are configured in every data center. Closet data center is used as service broker policy.

TABLE 1: MAIN CONFIGURATION PARAMETER

User Base	Region	Peak Hours (Local time)	Peak Hours (GMT)	Simultaneous Online Users During Peak Hrs	Simultaneous Online Users During Off-peak Hrs
UB1	0	7.00-9.00 pm	13:00-15:00	400,000	40,000
UB2	1	7.00-9.00 pm	15:00-17:00	100,000	10,000
UB3	2	7.00-9.00 pm	20:00-22:00	300,000	30,000
UB4	3	7.00-9.00 pm	01:00-03:00	150,000	15,000
UB5	4	7.00-9.00 pm	21:00-23:00	50,000	5,000
UB6	5	7.00-9.00 pm	09:00-11:00	80,000	8,000

TABLE 2: DATA CENTER CONFIGURATION PARAMETER

Parameters in CloudAnalyst	Values
VM Image Size	10000
VM Memory	1024 Mb
VM Bandwidth 1000	1000
Data Center-Architecture X86	X86
Data Center-OS	Linux
Data Center-VMM	Xen
Data Center-Number of Machines	5
Data Center-Memory Per Machine	2048 Mb
Data Center-Storage Per Machine	100000 Mb
Data Center-Available BW Per Machine	1000
Data Center-Number of Processors Per Machine	4
Data Center-Processor Speed	100 MIPS
Data Center-VM Policy	Time-Shared

The simulation result is shown in Fig. 3. Response to all user base (UB) by region is shown in Table 3. It indicated the average, minimum and maximum response time to all user bases.

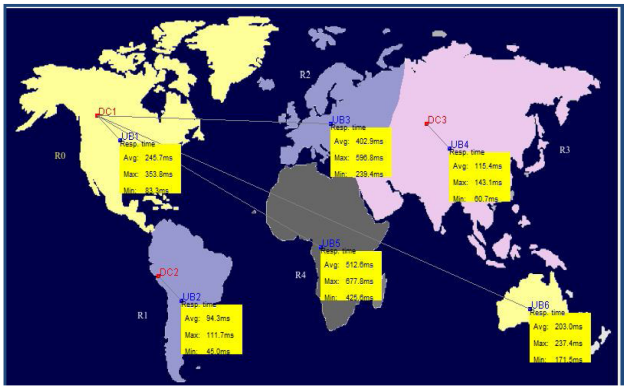

Fig. 3: Simulation Result for EAMLB with 3 DC

TABLE 3: RESPONSE TIME BY REGION FOR EAMLB IN CASE OF 3 DC

User Base	Avg(ms)	Min(ms)	Max(ms)
UB1	245.72	83.31	353.80
UB2	94.26	44.99	111.61
UB3	402.93	239.38	596.77
UB4	115.39	60.69	143.11
UB5	512.58	425.58	677.84
UB6	203.03	171.55	237.42

To assess the proposed algorithms, comparative analysis is done with existing load balancing algorithms in CloudAnalyst tool. The comparison of EAMLB algorithm is done with existing Round Robin and Active Monitoring and it is found that the average response time is better than both algorithms. Minimum response time is same as round robin and minimum than Active Monitoring Algorithm. Maximum response time is also minimum than both algorithm. The simulation result is shown Table 4.

TABLE 4: COMPARISON OF OVERALL RESPONSE TIME IN CASE OF 3 DC

Parameters	Load Balancing Algorithms		
	RR	AM	EAMLB
Avg (ms)	271.10	271.01	270.92
Min (ms)	44.94	44.99	44.94
Max (ms)	692.49	685.39	677.84

The graph is also drawn based on the average response time which shows the graphical representation in Fig. 4 that exactly clarify the downfall in response time by different algorithm.

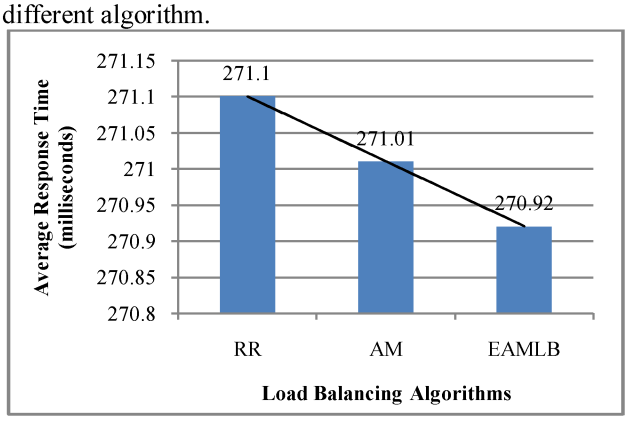

Fig. 4: Average Response Time by Different Algorithms in Case of 3 DC

Total virtual machine cost and data transfer cost is same for all three algorithms namely round robin, active monitoring and proposed algorithm. EAMLB achieves the objectives of this research work as defined in Introduction section and provide the better results as demonstrated in above result data and in graphical representation also. The EAMLB algorithm also equally distributes the load on different virtual machines as well as provides better response time.

VI. CONCLUSION AND FUTURE SCOPE

Proposed algorithm Enhanced Active Monitoring Load Balancing (EAMLB) algorithm is developed by enhancing the working of Active Monitoring algorithm. It improves the response time better than Round Robin and Active Monitoring.

The concept of proposed algorithm is also explained with the help of work flow diagram. For simulation of algorithm in virtual large scale cloud environment, graphical user interface based CloudAnalyst tool is used. Java language is used to develop the class file for implementation in tool.

Many soft computing approaches can be used to get better response time and virtual machines utilization in efficient manner. Different types of situations can be considered while developing an algorithm that will become helpful in providing better quality of service and increase the performance of cloud systems.

REFERENCES

[1] Mauch, Viktor, Kunze, Marcel, & Hillenbrand, Marius, "High performance cloud computing," Future Generation Computer Systems, International Journal of Grid Computing and eScience, Elsevier, ISSN: 0167-739X, Vol. 29, Issue. 2013, pp. 1408-1416, 2012.

[2] Singh, A.; Korupolu, M. & Mohapatra, D., "Server-storage virtualization: Integration and load balancing in data centers," International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, E-ISBN: 978-1-4244-2835-9, Print ISBN: 978-1-4244-2834-2, pp. 1-12, 2008.

[3] Junjie, Ni, Yuanqiang Huang; Zhongzhi Luan; Juncheng Zhang & Depei Qian, "Virtual machine mapping policy based on load balancing in private cloud environment," International Conference on Cloud and Service Computing (CSC), IEEE, E-ISBN: 978-1-4577-1636-2, Print ISBN: 978-1-4577-1635-5, pp. 292-295, 2011.

[4] Ahmed, Tanveer, & Singh, Yogendra, "Analytic Study Of Load Balancing Techniques Using Tool Cloud Analyst," International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol. 2, Issue. 2, pp. 1027-1030, 2012.

[5] Wickremasinghe, Bhathiya, " CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments," MEDC Project Report, Distributed Computing Project, CSSE Dept., University of Melbourne, pp.433-659, 2009.

[6] Wickremasinghe, Bhathiya, Calheiros, Rodrigo N., & Buyya, Rajkumar, "CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications," Proceedings of 24th International Conference on Advanced Information Networking and Applications (AINA), IEEE Computer Society, pp. 446-452, 2010.

[7] Brototi, Mondal, Dasgupta, Kousik & Dutta, Paramartha, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach," Elsevier, Procedia Technology, ISSN: 2212-0173, Vol. 4, pp. 783-789, 2012.

[8] Ajit, Mr. M., & Vidya, Ms. G., "VM Level Load Balancing in Cloud Environment," Proceedings of Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), IEEE, pp. 1-5, 2013.

[9] Mulay, Shreyas, & Jain, Sanjay, "Enhanced Equally Distributed Load Balancing Algorithm For Cloud Computing," International Journal of Research in Engineering and Technology, ISSN: 2319-1163, Vol. 2, Issue. 6, pp. 976-980, 2013.

[10] Babu, L.D. Dhinesh, Krishna, P. Venkata, "Honey Bee Behavior Inspired Load Balancing of Tasks in Cloud Computing Environments," Applied Soft Computing, Elsevier, ISSN: 2292-2303, Vol. 13, pp. 2292-2303, 2013.

[11] Kumar, B. Santosh, & Parthiban, Dr. Latha, "An Implementation of Load Balancing Policy for Virtual Machines Associated With a Data Center," International Journal of Computer Science & Engineering Technology (IJCSET), ISSN: 2229-3345, Vol. 5, Issue. 3, pp. 253-261, 2014.

[12] Rana, Madhurima, Bilgaiyan, Saurabh, & Kar, Utsav, "A Study on Load Balancing in Cloud Computing Environment Using Evolutionary and Swarm Based Algorithms," Proceedings of International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), IEEE, pp. 245-250, 2014.

[13] Page, Andrew J. and Naughton, Thomas J., "Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing," Proceedings of 19th International Conference on Parallel and Distributed Processing Symposium, IEEE, pp. 189a, 2005.

[14] Dasgupta, Kousik, Mandal, Brototi, Dutta, Paramartha, Mondal, Jyotsna Kumar, & Dam Santanu, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing," Proceedings of 1st International Conference on Computational Intelligence: Modelling Techniques and Applications, Elsevier, pp. 340-347, 2013.

[15] Behal, Veerawali, & Kumar, Anil, "Cloud Computing: Performance Analysis of Load Balancing Algorithms in Cloud Heterogeneous Environment," Proceedings of 5th International Conference- Confluence The Next Generation Information Technology Summit (Confluence), IEEE, pp. 200-205, 2014.

[16] Ghutke, Bhushan, & Shrawankar, Urmila, "Pros and Cons of Load Balancing Algorithms for Cloud Computing," Proceedings of International Conference on Information Systems and Computer Networks, IEEE, pp. 123-127, 2014.

