2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015)
Dehradun, India, 4-5 September 2015

Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing

Shikha Garg*, Rakesh Kumar Dwivedi† and Himanshu Chauhan‡

* Uttarakhand Technical University Dehradun, India shikha.incoer@gmail.com
†Teerthanker Mahaveer University Moradabad, India dwivedi.rakesh02@gmail.com
‡ College of Engineering Roorkee Roorkee, India himanshuchauhan@coer.ac.in

***Abstract*—Cloud computing is an emerging area that offers many potential benefits to various organizations and common user. It is the extended form of distributed computing. . It is based on on-demand-service model in which information, software, infrastructure and other services are provided as per the client requirement at some instance of time. Load balancing is used to distribute workload among multiple cloud systems or nodes to get better resource utilization. It is the prominent means to achieve efficient resource sharing and utilization. Load balancing has become a challenge issue now in cloud computing systems. To meets the users huge number of demands, there is a need of distributed solution because practically it is not always possible or cost efficient to handle one or more idle services. Servers cannot be assigned to particular clients individually. Cloud Computing comprises of a large network and components that are present throughout a wide area. Hence, there is a need of load balancing on its different servers or virtual machines. This research paper proposed an algorithm that focuses on load balancing to reduce the situation of overload or under load on virtual machines that leads to improve the performance of cloud substantially. Comparative analysis is being done with the help of CloudAnalyst tool.**

Keywords: cloud computing; cloudlet; CloudAnalyst; resource utilization; synchronized; virtaul machine.

I. INTRODUCTION

Cloud computing is referred as a type of Internet based Computing. In cloud computing, cloud can be viewed as Internet. It provides accessibility of resources on Internet for its entire user and it also offers computing functionality on demand as a service that can be free or at cost. Customer uses these services according to pay-as-you-use model. Cloud computing aims to share data, resources and services among its users. Cloud computing provides infrastructure, software and platform as a service for different task that are submitted by the user. Software as a Service standard offers application software functionality within a cloud computing environment. Platform as a Service is defined as combination of software and product development tools. PaaS services are hosted by CSPs infrastructure. User can design and develop applications on providers platform . Infrastructure as a Service refers the availability of hardware resources to the user. With this type of service, customer can utilize resources including storage, memory space, network equipment, virtual machines and server space [1, 2]. The development of load balancing algorithms in cloud computing is still an open research area. There is a need to explore and implement existing load balancing algorithms that may improve the working ability of cloud computing systems [2, 3]. Resource utilization is one of the performance parameter that refers the degree to which resources are utilized. Maximum utilization of resources is provided by the efficient load balancing algorithm. Hence, the proposed algorithm provides maximum utilization of each virtual machines that are created by data center in cloud systems. CloudAnalyst tool is used for simulation and analysis of proposed algorithm.

II. RELATED WORK

Related work that focuses on the load balancing in cloud environment is described in this section. In [3] a dynamic load balancing mechanism is proposed for open cloud computing federation (OOCF) is proposed. It improves various aspects of Ant Colony Algorithm (ACO) proposed in [4]. Authors showed with the help of simulation that this algorithm can effectively manage the heavy workload on distributed cloud systems. However better results can be achieved by slight changes in parameters of ACO. In [5] optimal resource allocation in dynamic cloud system is achieved by Load Balancing Ant Colony Optimization (LBACO) algorithm. The authors showed with the help of simulation that LABCO outperforms the existing algorithms FCFS (First Come First Serve) and the basic ACO (Ant Colony Optimization) proposed in [6]. With simulation it is demonstrated that this algorithm is able to balance load of entire system effectively that leads to better resource utilization. In [7] contributed a load balancing algorithm which helps to minimize allocation time of user request and minimize the system overhead. They used the concept of virtual machine (VM) migration form overloaded VM to other in order to maintain efficient resource utilization. In [8] an Active Monitoring Load Balancer Algorithm is proposed. It maintains the information about each VMs and number of request currently allocated to which VM. This algorithm returns the VM id to Data Center Controller (DCC) the request is assigned to that VM identified by that id. The proposed algorithm may lead to over/under utilization of VMs/ resources. Authors in [9] proposed a novel VM- assign algorithm for efficient allocation of incoming job on virtual machine existing in cloud computing systems. The presented algorithm focus on finding out least loaded virtual machine and then incoming jobs are allocated on them intelligently. Authors

showed with the help of simulation that the proposed algorithm outperforms the Active VM-load balancer algorithm proposed in [8] and solves the problem of inefficient utilization of VMs / resources. In [10] authors proposed Central Load Balancer (CLB) technique in which the situation of over loading and under loading of virtual machines is avoided. CLB assigns the load to various virtual machines corresponding to their priority and states. Authors explained using simulation that CLB technique based load balancing algorithm performs better than Round Robin (RR) and Throttled algorithm. However their presented algorithm is lacking in consideration of current resource utilization including processor and memory with which help load distribution can be more robust and dynamic.

III. PROPOSED SYNCHRONIZED THROTTLED LOAD BALANCING ALGORITHM

The solution have been proposed for efficient utilization of virtual machines using Synchronized Throttled Load Balancing (STVMLB) algorithm. In cloud computing environment, it is essential requirement to balance the load on virtual machines which are created on data centers. To avoid the situation of overload or under load, there should be a proper balancing of load on machines so that performance of system can be increased. Proposed technique is a better attempt to maximize the utilization of resources including virtual machines. STVMLB synchronizes all virtual machines in better manner. This technique has the extended concept of Throttled Algorithm. In this technique, the following notations as been used:
j1, j2,.,jn : denotes jobs
UB: Set of jobs known as user base
v1, v2,.,vn :different virtual machines
vmid : Virtual machine which has to be checked
VM : Virtual machine which defines the state either AVAILABE or BUSY
currVm : Virtual machine which has to be checked for availability
Hash map is used to store the vmid and its corresponding state either AVAILABLE or BUSY. In the beginning, all VMs state is available. When Data Center Controller (DCC) receives a new request, it asks SynchronizedThrottledVmLoadBalancer for job allocation. If all VM has been efficiently allocated, then load balancer starts the checking of availability form first VM, if all VMs are not allocated then value of currVm will be incremented for checking the availability of next VM. The vmid is returned to the DCC which notifies to the SynchronizedThrottledVmLoadBalancer to change the values in allocation table and keep the state BUSY of allotted machine in vmStateList. When VM finishes the job processing and DCC receives the response cloudlet then DCC notifies to STVMLB for deallocation of VM and then SynchronizedThrottledVmLoadBalancer change the state of VM as AVAILABLE. If more jobs are waiting then process of allocation is started from next VM until the last VM. In Throttled, every time the vmStateList is checked from first index and that makes some VM overloaded or underloaded. Fig. 1 represents the workflow of proposed algorithm.
Algorithm: SynchronizedThrottledVmLoadBalancer
Input: Jobs j1, j2,.,jn to be assigned to virtual machine, virtual machines v1,v2,.,vn available at Data Center
Output: All submitted jobs j1, j2,.,jn are allocated one by one to each available virtual machine

TABLE I. APPROXIMATE DISTRIBUTION OF FB USER

Regions Divided in GUI Screen	Region Id in CloudAnalyst	Number of Users (Million)
North America	0 (R_0)	80
South America	1 (R_1)	20
Europe	2 (R_2)	60
Asia	3 (R_3)	27
Africa	4 (R_4)	5
Ocenia	5 (R_5)	8

1) currVm = -1
2) Create a Hash Map vmStateList¡vmid, VirtualMachineState¿
3) DCC receives a new request
4) DCC queries the SynchronizedThrottledVmLoadBalancer for next job allocation **/* SynchronizedThrottledVmLoadBalancer starts checking for the availability of VM from the very first VM and later from next to allocated VM */**
5) INCREMENT currVm
6) IF currVm >= vmStatesList.size() THEN
7) currVm = 0
8) END IF
9) IF vmStatesList.size() > 0 THEN
10) FOR each VM next to the recently allocated VM
11) IF VM state is AVAILABLE THEN
12) currVm = index value of checked VM
13) END IF
14) END FOR
15) Allocate the VM having id currVm
16) Return currVm to the DCC
17) DCC notifies SynchronizedThrottledVmLoadBalancer to update allocation table accordingly
18) ELSE
19) currVm = -1
20) Return currVm to the DCC
21) ENDIF
22) IF VM finishes the job processing and DCC receives the response cloudlet THEN
23) DCC indicates SynchronizedThrottledVmLoadBalancer for VM deallocation
24) ENDIF
25) IF more request/job exist THEN
26) Repeat from Step 3
27) ENDIF

IV. EXPERIMENTAL SETUP IN CLOUDANALYST TOOL

Fig.2 represents the architecture of CloudAnalyst tool. The development of CloudAnalyst is done on top of CloudSim toolkit. It extends the functionalities of CloudSim. Internet and Internet Application behavior is modeled in its concept. With its help, a modeler can focus on the simulation complexities rather than spending too much time on the technicalities of programming [11, 12].

For this proposed work, the data from Facebook, a social

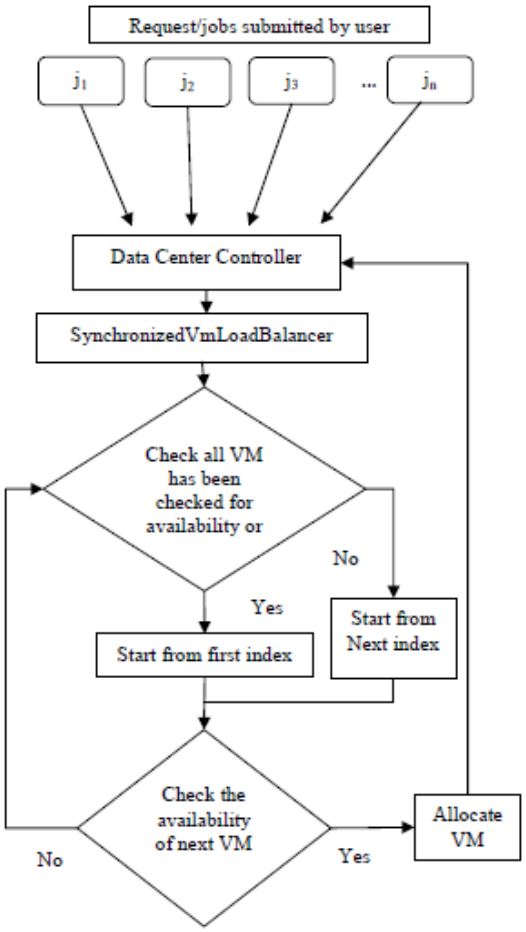

Fig. 1. Workflow of STVMLB.

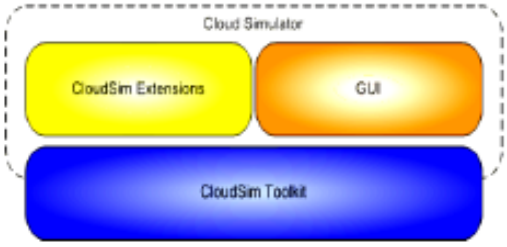

Fig. 2. Architecture of the Cloudanalyst (Wickremasinghe, 2009).

networking site is used. It is a large scaled Internet application among other social networking applications. It approximately has over 200 million users who are registered in this site worldwide. On the date of 18 Jun, 2009, the approximate distribution of the users of this site is given in Table I along with the regions overall the world [11]. For simulation purpose, similar system is assumed but it is 1/10th of scale of the given data.

For simulation of proposed algorithm, assumed parameters for main configuration are depicted in Table II. One user base is defined in each and every region of the world. It is assumed that most of the user uses the social networking application

TABLE II. 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

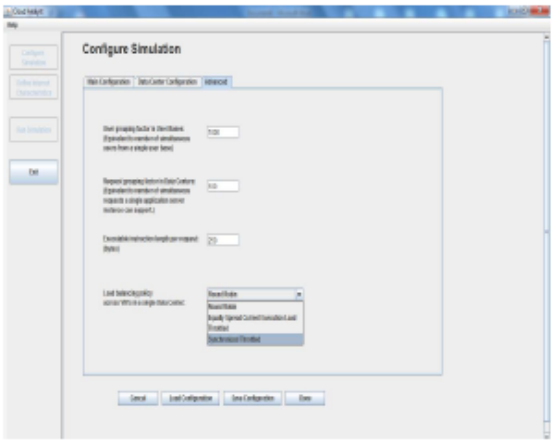

Fig. 3. Configure Simulation Panel in CloudAnalyst Tool.

for two hours in evening. During the peak timings, only five percent of users are online along with 1/10th users are in offpeak hours. Time is depicted in Table II according to the particular region. It is also assumed that when user is online then he/she is sending new request in next 5 minutes. Closet Data Center is used as service broker policy.

V. RESULT AND ANALYSIS

Synchronized Throttled VM Load Balancing (STVMLB) Algorithm is implemented in Java programming 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. This file is added to the GUI part of configure simulation is CloudAnalyst tool as shown in Fig. 3.

In first case, five VM are used with a single data center for configuration in CloudAnalyst tool. Table III denotes that Synchronized Throttled VM Load Balancing (STVMLB) algorithm provides better utilization of virtual machines rather than Active Monitoring and Throttled algorithms. As the proposed algorithm is designed after some modification in Throttled algorithm, it almost perfectly uses the virtual machines rather than Throttled. In Throttled, every time the availability of VM is checked from the very first VM but in STVMLB the availability checking starts from the next to allocated VM. So by using STVMLB, some VM will escape from the condition of overload and under load. Round Robin allocates the virtual machines better than proposed algorithm but STVMLB also

TABLE III. USE OF VIRTUAL MACHINES (IN NUMBERS) IN CASE OF 1 DC HAVING 5 VM

VM	Load Balancing Algorithms			
	RR	*AM*	*Throttled*	*STVMLB*
0	24205	33034	14852	24106
1	24205	23322	12473	24079
2	24204	20690	12285	24150
3	24204	23015	12151	24273
4	24204	20961	12026	24414

TABLE IV. USE OF VIRTUAL MACHINES (IN NUMBERS) IN CASE OF 1 DC HAVING 25 VM

VM	Load Balancing Algorithms			
	RR	*AM*	*Throttled*	*STVMLB*
0	4842	22312	8923	4701
1	4842	5010	2967	4674
2	4842	4022	2805	4680
3	4842	3787	2763	4662
4	4842	4778	2881	4671
5	4842	5758	3165	4665
6	4842	3487	2722	4668
7	4842	4638	2815	4692
8	4842	4316	2736	4691
9	4842	4055	2688	4696
10	4842	6796	2930	4714
11	4842	3662	2544	4731
12	4842	3665	2594	4732
13	4842	4188	2507	4762
14	4842	3594	2433	4802
15	4842	4758	2606	4820
16	4842	4409	2470	4855
17	4841	3844	2428	4898
18	4841	3539	2430	4922
19	4841	3020	2353	4974
20	4841	4776	2293	5052
21	4841	3605	2281	5111
22	4841	2852	2240	5207
23	4841	3658	2215	5270
24	4841	2513	2187	5392

checks the state of VM either it is AVAILABLE or BUSY which is not checked in Round Robin. In consideration of response time, the STVMLB gives better average response time than Round Robin but it is not better than Active Monitoring and Throttled load balancing algorithm.

In second case, twenty five virtual machines with single data center are configured in CloudAnalyst tool. After simulation, it is found that STVMLB almost solve the problem of over utilization and under utilization of virtual machine. The results are shown in Table IV. STVMLB better utilizes the VM than Active Monitoring and Throttled algorithm. Still RR allocates the virtual machine in case of 25 VM better than STVMLB. After getting simulation result, it is found that the average response time of STVMLB is better than Round Robin and Active Monitoring algorithm. However, average response time is worse than Throttled load balancing algorithm.

VI. CONCLUSION AND FUTURE SCOPE

The proposed algorithm STVMLB is designed by making modification in the concept of Throttled load balancing algorithm. This proposed algorithm increases the utilization of VM better than Throttled and Active Monitoring. Work flow of algorithm is shown with the help of diagram which is easier to understand the concept of algorithm. CloudAnalyst tool is used for simulation purpose which is GUI based software. Proposed algorithm is implemented in Java language and tested in large scale cloud environment. Proposed dynamic load balancing algorithm produces better result as to efficiently allocate the coming request and maximize the response time in cloud environment. Though response time is not better than Throttled algorithm, it will be improved in our future work. Load balancing algorithm can also be designed with other approaches like soft computing that will efficiently utilize virtual machines as well as it will minimize the response time. Geographically locations of the task, resources and Quality of Service (QoS) parameters can be considered. Paradigm of parallel computing and high performance computing can be used to maximize the performance of load balancing algorithms.

REFERENCES

[1] Rimal, B. P., Eunmi Choi Lumb, I., A Taxonomy and Survey of Cloud Computing Systems, Fifth International Joint Conference on INC, IMS and IDC, IEEE, E-ISBN: 978-0-7695-3769-6, Print ISBN: 978-1-4244-5209-5, pp. 44-51, 2009.

[2] Nuaimi, K. A.; Mohamed, N.; Nuaimi, M. A. Al-Jaroodi, J., A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms, Second Symposium on Network Cloud Computing and Applications (NCCA), IEEE, Print ISBN: 978-1-4673-5581-0, pp. 137- 142, 2012.

[3] Zhang, Zehua Zhang, Xuejie, A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation, Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), IEEE, pp. 240-243, 2010.

[4] Schoonderwoerd, Ruud, Holland, Owen, Bruten, Janet, Ant-like agents for load balancing in telecommunications networks, Proceedings of Agents, Marina del Rey, pp. 209-216, 1997.

[5] Li, Kun, Xu, Gaochao, Zhao, Guangyu, Dong, Yushuang, Wang, Dan, Cloud Task scheduling based on Load Balancing Ant Colony Optimization, Proceedings of 6th Annual ChinaGrid Conference, IEEE, pp.3-9,2011.

[6] Ku-Mahamud, K. R., Nasir, Husna Jamal Abdul, Ant Colony Algorithm for Job Scheduling in Grid Computing, Proceedings of Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation (AMS), IEEE, pp. 40-45, 2010.

[7] Ren, Haozheng, Lan, Yihua, Yin, Chao, The Load Balancing Algorithm in Cloud Computing Environment, Proceedings of 2nd International Conference on Computer Science and Network Technology, IEEE, pp. 925-928, 2012.

[8] Mahalle, Hemant S., Kaveri, Parag R. Chavan, Vinay, Load Balancing On Cloud Data Centres, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN: 2277-128x, vol. 3, Issue. 1, pp. 1-4, 2013.

[9] Damanal, Shridhar G. Reddy, G. Ram Mahana, Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines, Proceedings of 6th International Conference on Communication Systems and Networks (COMSNETS), IEEE, pp. 1-4, 2014.

[10] Soni, Gulshan, Kalra, Mala, A Novel Approach for Load Balancing in Cloud Data Center, Proceedings of International Advance Computing Conference (IACC), IEEE, pp. 807-812, 2014.

[11] 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.

[12] 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.