International Journal of Research in Advent Technology, Vol.6, No.5, May 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

An Improved Regulated Load Balancing Approach for Cloud Computing Environment

Devika Singh[1], Ashish Vashisht
Computer Science & Engineering[1, 2], M. Tech. Student[1], Assistant Professor & Head[2]
Email: devikasingh711@gmail.com[1],ashish.vashisht@kitm.in [2]

Abstract-With the advancement in technology, use of cloud computing is increasing day by day. Due to specific characteristics like pay as per usage and use anywhere, any time without human intervention has increase its user exponentially. Emergence of cloud computing has reduced the startup cost from service provide point of view. This has also results in low cost services for service users. Cloud computing has inherited some features from distributed, parallel and grid computing. During inheriting good features from its ancestor technologies, cloud computing has also inherited the associated issues. Performance and security are the major hurdles in wide spread acceptance of cloud computing. In this paper, author has proposed the enhanced version of existing regulated load balancing approach for cloud computing. To check the performance of proposed approach, author has used the cloud analyst simulator. Through simulation analysis, it has been found that proposed improved version of regulated load balancing approach has shown better performance in terms of cost, response time and data processing time.

Index Terms-Cloud Computing, Load Balancing, Cloud Analyst, Task Scheduling

1. INTRODUCTION

When computing services are provided through internet, then this model of delivery of services is called cloud computing. The cloud offers services like SaaS-Software-as-a-Service, PaaS –Platform as a Service and IaaS-Infrastructure-as-a-Service. Cloud computing is the consequence of evolution through parallel, distributed, cluster, grid and utility. Cloud provides some unique facilities like pay per usage, use any time anywhere at efficient cost. This model can be implemented in public, private and hybrid model. This concept has turned into reality due to concept of virtualization. Security and load balancing are the major issues in the widespread acceptance of cloud computing. Figure 1shows the basic concept of cloud computing [1, 2, 3].

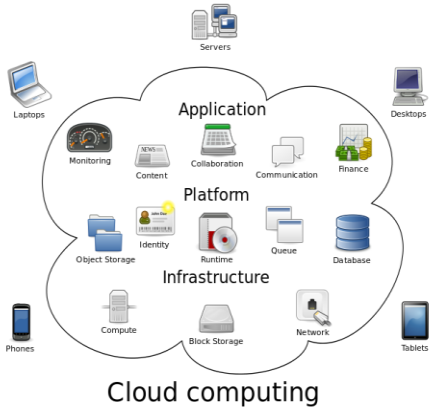


Fig. 1. Outline of Cloud Computing [1]

Organization of work is as follows: Section 2 describe the basic of load balancing. Load balancing types are explained in section 3.Section 4 covers the related work in the area of cloud computing load balancing. Proposed load balancing algorithm has been described in section 5. Section 6 covers the details of simulator. Details of simulation setup are given in section 7. Results and its analysis has been discussed in section 8. Conclusion and future work scope has been given in section 9.

2. LOAD BALACING

Load balancing is define as the distribution of resources, simultaneous working of the schedulers, efficiency enhancement, and minimization of response time via a suitable matching of job to the available resource. Simultaneous working of the schedulers involves the distribution of load in equal manner among the processors. To restore the balance dynamic load balancing also known as load sharing or load migration is employed [4].

It is done by distributing the entire load to the individual processors of the complete structure for obtaining efficient resource mapping and concurrently removing the possibility of overloading or under loading of the nodes in the network. It is done to achieve for better ratio of user realization and resource utilization, thereby enhancing the throughput of the complete system. If done in proper manner the load management can limit the consumption of the available. It also helps in executing failures, making the system scalable, and over-burdening, minimizing

response time etc. The main goals of the load balancing algorithms are listed below [5]:

- Rate Efficiency: Load balancing support should execute with lesser rate in the given framework. To make certain the excessive efficiency, reduced response time, and confine the overload.
- Scalability: The structure for which load balancing methods are executed should be capable of being altered in elements in later time.
- Elasticity: The practical requirements should be able to manage with circumstances; they should be elastic and adaptable. To compel prospect variations in the system.
- Priority: The resources and the tasks should be arranged as per priority. So higher works show signs of upgrading and execution.
- Resource utilization: To make sure that the available resources are being utilized in a perfect manner.
- Backup: In case of failure of the structure, load balancing algorithms must have a backup schedule.
- Homogeneous nature: To treat all tasks in the system homogeneously irrespective of their source [6, 7].

3. LOAD BALANCING ALGORITHM TYPE

If load balancing policy is initiated by the sender then it is called sender initiated load balancing policy. If receiver initiates it then it is called receiver initiated load balancing policy [8].
If concept of load balancing is predefined and not changing according to situation then it is called static load balancing algorithm. If load balancing approach is changing itself according to situation, then it is called dynamic load balancing approach. All load balancing algorithms types are collectively shown in figure 2 [9, 10].

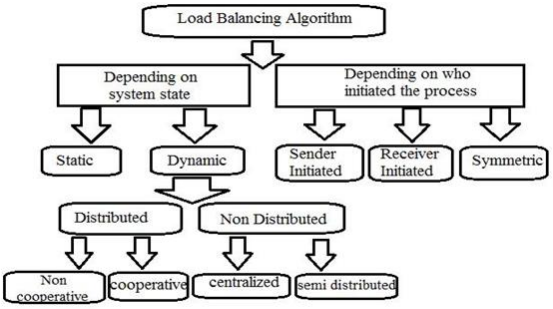


Fig. 2. Load balancing algorithm types [9]

4. RELATED WORK

In distributed computing, managing the load is essential so that all the virtual machines are provided with equal number of tasks. It supports in achieving client satisfaction and optimum utilization of resources by assuring an efficient and unbiased sharing of resources among the VMs. Main features of balancing the load includes the reduction in using resources and faults. It also allows in evading jams, etc. The underlying section deals with systematic analysis of existing load balancing techniques considering various parameters in order to reduce the overhead, response time and improve the performance, etc.

- N. Swarnkar et al. in [11] have described a conventional round robin approach for balancing the load. A group of available VMs gets the tasks on the random basis and the process of task allocation continues in circular (round) motion. When a task is mapped to the VM then it goes to the last position in the VM list. The discussed approach doesn't have any idea of size of the incoming tasks so suffers with the disadvantage of some overloaded nodes. Besides this, the benefit of this algorithm is that inter-process communication is not required.
- S. Moharana et al. in [12] have presented a Weighted Round Robin approach for balancing the load in cloud environment. The described scheme is a combination of weight assigning and round robin approach. The capacity of the VM to accommodate the tasks helps in assigning weight to the VM and after selecting the VM conventional round robin approach is executed.
- K. Mahajan et al. in [13] have described a Round Robin approach with Server Affinity to improve the response time. The discussed approach maintains two data structures whose details are given as under:

  1) Hash map: The details of the last task assigned to VM is stored in harsh map.

  2) VM state list: The state of each VM, either it is idle or busy is saved in VM state list

  For all the incoming tasks the scheduler checks the entry of the task in the harsh map and assigns the task to the idle VM without running the round robin algorithm.
- A. Makroo et al. in [14] have discussed basic Throttled load balancing approach for cloud environments in which it was considered that VM has the capacity to handle single task only and the incoming tasks are assigned to the idle Vm's which are selected randomly if more VMs are found to be idle.

- S.G. Donamal in [15] has discussed the Modified Throttled algorithm to reduce the response time by maintaining a table of contents that contains the record and status of VMs. The assigning of tasks to the VM doesn't start from the beginning as in usual throttled approach but the mapping is done smartly by assigning it to VM next to already assigned VM.
- Meenakshi Sharma et al. in [16] have discussed active VM load balancing approach. In this approach, information about the load assigned to each node is maintained & whenever a new task arrives, it is mapped with the least loaded machine. Due to the consideration of VM present load during mapping, it is somewhat dynamic in nature.
- Meenakshi Sharma et al. in [17] have given a new approach for load balancing algorithm in a cloud environment. The algorithm maintains the expected response time corresponding to each VM. It is estimated on the basis of load assigned to each VM. Whenever a new request arrives, it maps it with that VM whose response time is least.
- Jasmin James et al. in [18] descried the improvised version of basic active VM load balancing approach named as weighted active monitoring approach. For load balancing the said approach examines the number of jobs assigned to the VM and its capacity to handle the tasks before mapping of new task to VM. When the new task arrives, it is assigned to the VM having high power and low weight.
- M. M. Ladani et al. in [19] presented an active load balancing algorithm that assigns the jobs to the virtual machine having highest power. The available VM having higher weight count means having maximum allocation of resources is selected for the task.
- M. Vaidehi et al. [20] in have presented a technique to balance the load among the overloaded and under-loaded nodes by simply shifting the jobs from overloaded node to under-loaded node in case any virtual machine is found overloaded. To accomplish this task the technique tracks the data of VM id, task id and no of active tasks allocated to the VM.
- A. Jain et al. in [21] have proposed a hybrid distributed load balancing approach by merging the best features of join idle queue, minimum completion time and join shortest queue. Moreover authors has also embedded a prior load checking logic to avoid the overloading situation,

5. PROPOSED ALGORITHM

In the basic regulated approach, merely one job can be mapped to a virtual machine and there is no parameter of virtual machine selection. Virtual machine which is free can be mapped to job. During mapping, basic regulated approach does not pay attention to capacity of virtual machine and size of task. Therefore, association between job and machine is not good. This results in overloading of some virtual machines and delay in processing of some jobs.
However, in the proposed improved regulated approach, weight is assigned to all virtual machine. Based on the available resources allotted to virtual machine, jobs count and size in the queue of that virtual machine, weight of each virtual machine is calculated. Improved regulated dispatcher will choose that virtual machine whose weight is largest among the available group of virtual machine. A weighted peak level is maintained to escape the overloading situation. If the weight of virtual machine is less than peak level then job is not assigned to that machine. Dispatcher manages the virtual machine mapping table. This table stores the identification of each virtual machine along with its resources information, weight, and number of jobs allotted and their size. Comparative analysis of basic regulated and improved regulated approach has been shown in table I.

Basic Regulated	Improved Regulated
There is no queue at virtual machine level.	There is provision of queue at virtual machine level.
Virtual machine can entertain one job at a time.	Depending upon the weight of virtual machine and peak value, machine can entertain multiple jobs at a time.
Machines are selected on random basis.	Machines are selected based upon their weight.

Table I: Comparison of Basic Regulated and Improved Regulated

Algorithm Improved Regulated ( )
{
$M= \{M_1, M_2, \ldots\ldots M_n\}$
//Available set of virtual machines
for (j=1 to n)
{
// Capacity of machine in the starting
C_j= Capacity (M_j)
// Weight of machine in the starting
$W_j=C_j$

```
        }
        // Peak value at time t
        P(t)= Least (Wi) at time t;
        While (Job is there in queue of Improved
        Regulated dispatcher)
        {
            Select the current job from the job queue
            of the dispatcher.
            Select the machine having highest weight.
            Discard those machine whose weight is
            less than peak value
            Assign the selected job to selected
            machine.
            Recalculate the weight of each machine.
        for (j=1 to n)
        {
            If (there is at least one job is associated to
            machine)
            {
                    Wj= Cj / ∑k Sk;
            }
        }
        Recalculate the peak value
    }
}
```

6. SIMULATOR DETAIL

B. Wickremasinghe et al. gave a command based cloud simulator that was an extension of the Cloudsim for the cloud analysts. The structure of cloud analyst is shown in figure 3.

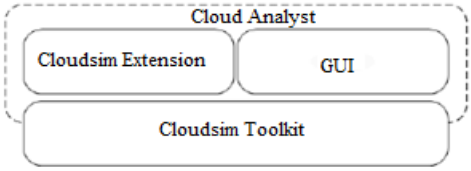


Fig. 3. Cloud analyst structure [22]

Key mechanism of the cloud simulator is given as under:

Fig. 4. Cloud Analyst GUI [22]

- GUI Package: It generates and maintains the interface that helps the user to organize the surroundings easily. The configuration environment of cloud analyst that is based on GUI is shown in figure 4.
-
- User Base: According to simulation configuration requirements, the user base generates the virtual traffic from different physical areas.
- Simulation: The parameters for experimental setup are handled by this element.
- Internet: The internet is tested experimentally by this element.
- Internet Characteristics: The internet characteristics of parameters for experimental setup are handled by this element.
- Cloud Application Service Broker: The routing of traffic between the user base and data center is managed by this aspect.
- Data Center Controller: the jobs of the data center are controlled by this element.
- VM Load Balancer: Various load balancing policies that users have configured are tested experimentally by this factor [22].

7. SIMULATION SETUP

Simulation State delivers the facts about the numerous factors on which proposed approach has been analyzed. Simulation state also delivers the facts about the network structure and available resources. A simulation module has been realized in Java language using the cloud analyst simulator. An imaginary assumption has been done in cloud analyst simulator. Corresponding to major part of the earth, world is divided into six regions. Corresponding to 6 continents of world, 6 user bases are modelled. This has been shown in the figure 4.

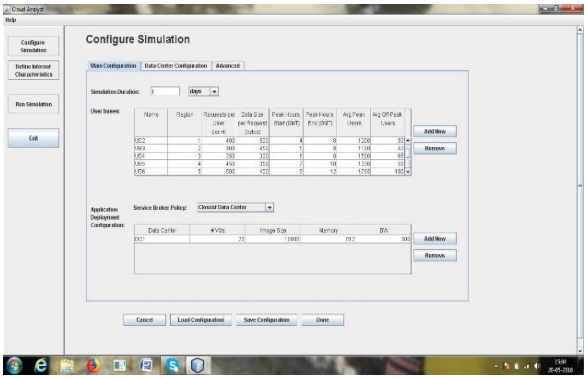


Fig. 5. Configuration of user bases [22]

Different user bases are in different time zones and due to this peak hours of different user bases will be different. Simulation was conducted for one day. Service broker policy was closest data center. Maximum number of machines which can be created on a data center are 20. Figure 5 shows the details of all the factors while tested the proposed approach. User base is used to generate the required traffic and the request for the services. Parameters for this include Region, request/hour, Data size, Peak hours, Average number of users present in peak hours and Avg. Off-Peak users.

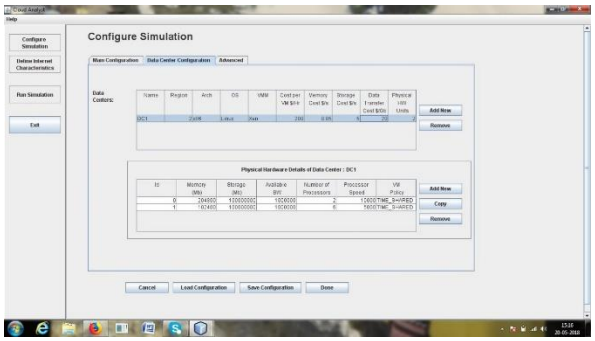


Fig. 6. Configuration of Data Centers [22]

Figure 6 shows the configuration of various data center parameter. Datacenter lists the available resources including hardware, their availability in a region, Architecture, Operating system, VMM, Cost and physical hardware units. In this work, only one data center has been chosen. This data center lies in the region 2. It has two hosts. Host 1 has two processors while host 2 has 6 processors. Fixed amount of primary, secondary, processing speed and bandwidth is allocated to them. Cost of using virtual machine, using primary memory, secondary memory and data transfer cost is fixed during data center configuration. Architecture, operating system and virtual machine manager are also selected during data center configuration.

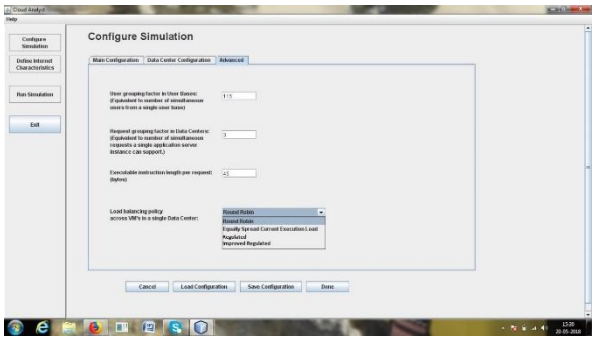
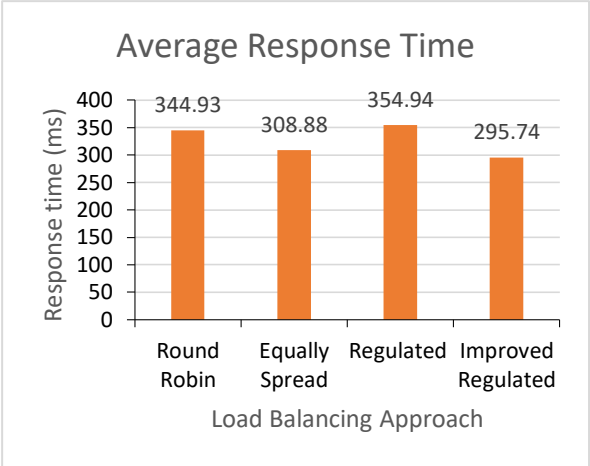


Fig. 7. Other parameters for simulation [22]

Figure 7 shows the setting of some other parameters. Around 115 number of users from a single user base can access the services simultaneously. An application server can support at most three simultaneous requests.



8. RESULTS & ANALYSIS

To analyze the performance of proposed improved regulated approach, results on different parameters has been shown graphically in figure 8, 9and 10 respectively.

Fig. 8. Average Response Time Comparison

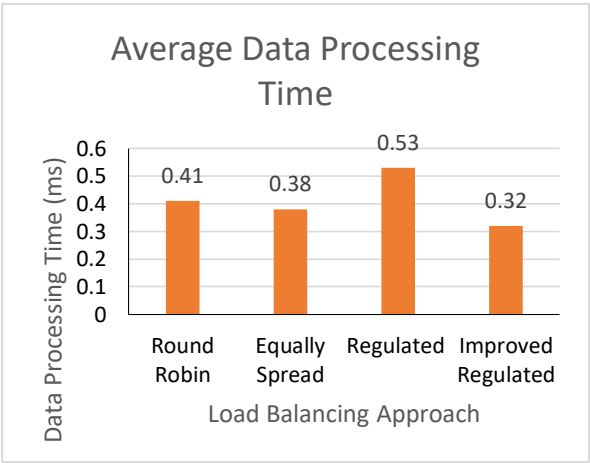


Fig. 9. Average Data Processing Time Comparison

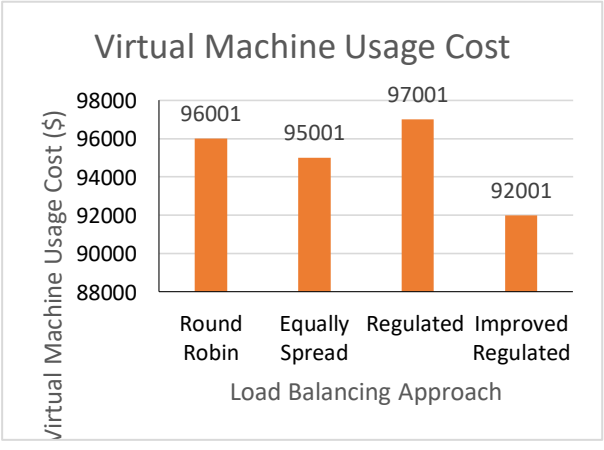


Fig. 10. Virtual Machine Usage Cost Comparison

9. CONCLUSION

Response time, processing time, recovery time, cost and service level agreement are the key parameters on which we judge the performance of cloud computing environment. All the mentioned performance parameter can be achieved in satisfactory manner by using a suitable load balancing approach. In this thesis, improved regulated algorithm (IRA) has produced the better result relative to equally spread, throttled and round robin approach on all the performance-oriented parameters. The proposed improved regulated load balancing approach is centralized in nature. Merging this with some other approach, so that resultant approach becomes distributed in nature, will make the proposed approach ideal for cloud environment.

REFERENCES

[1] Jain, A.; Kumar, R. (2014): A taxonomy of cloud computing. International journal of scientific and research publications, 4(7), pp. 1-5.

[2] Buyya R.; Yeo C.S.; Venugopal S.; Broberg J.; Brandic I.(2009): Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5thutility. Future Generation computer systems, 25(6), pp. 599-616.

[3] Sasikala P. (2011): Cloud computing: Present status and future implications. International Journal of Cloud Computing, 1(1), pp. 23-36.

[4] Jain, A.; Kumar, R. (2016): A comparative analysis of task scheduling approaches for cloud environment. 3rdInternational Conference on Computing for Sustainable Global Development (INDIACom), pp. 1787-1792.

[5] Florence A. P.; Shanthi V. (2013): Intelligent Dynamic Load Balancing Approach for Computational Cloud, International Journal of Computer Applications, pp. 15-18.

[6] Alakeel A.M. (2010): A guide to dynamic load balancing in distributed computer systems, International Journal of Computer Science and Information Security, 10(6): pp. 153-160.

[7] Jain, A.; Kumar, R. (2017): Critical analysis of load balancing strategies for cloud environment. International Journal of Communication Networks and Distributed Systems, 18(3-4), pp.213-234.

[8] Khiyaita A.; El Bakkali H.; Zbakh M.;, El Kettani D. (2012): Load balancing cloud computing: state of art, In National Days of Network Security and Systems (JNS2), pp. 106-109.

[9] Al Nuaimi K.; Mohamed N.; Al Nuaimi M.; Al-Jaroodi J. (2012): A survey of load balancing in cloud computing: Challenges and algorithms, In Second Symposium on Network Cloud Computing and Applications (NCCA), pp. 137-142.

[10] Jain, A.; Kumar, R. (2016) Scalable and trustworthy load balancing technique for cloud environment. International Journal of Engineering and Technology, 8(2), pp. 1245-1251.

[11] Swarnkar N.; Singh A. K.; Shankar (2013): A Survey of Load Balancing Technique in Cloud Computing, International Journal of Engineering Research & Technology, 2(8), pp. 800-804.

[12] Moharana S. S.; Ramesh R. D.; Powar D. (2013): Analysis of Load Balancers in Cloud Computing, International Journal of Computer Science and Engineering (IJCSE), 2(2), pp. 101-108.

[13] Mahajan K.; Makroo A.; Dahiya D. (2013): Round Robin with Server Affinity: A VM Load Balancing Algorithm for Cloud Based Infrastructure, Journal of Information Process System, 9(3), pp. 379-394.

[14] Makroo A.; Dahiya D. (2014): An efficient VM load balancer for Cloud. Applied Mathematics, Computational Science and Engineering, pp. 333-359.

[15] Domanal S.G.; Reddy G.R.M.(2013); Load Balancing in Cloud Computing using Modified Throttled Algorithm, in IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 1-5.

[16] Sharma M.; Sharma P. (2012): Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm, International Journal of Advanced Computer Science and Applications, 3(2), pp. 86-88.

[17] Sharma M.; Sharma P.; Sharma S. (2012): Efficient Load Balancing Algorithm in VM Cloud Environment, International Journal of Computer Science and Technology, 3(1), pp. 439-441.

[18] James J.; Verma B. (2012): Efficient VM Load Balancing Algorithm for a Cloud Computing Environment, International Journal on Computer Science & Engineering, 4(9), pp. 1658-1663.

[19] Ladani M. M.; Gupta V. K. (2013): A Framework for Performance Analysis of Computing Clouds, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2(6), pp. 245-247.

[20] Vaidehi M.; Rashmi K.S.; Suma, V. (2012): Enhanced Load Balancing to Avoid Deadlock in Cloud, Special Issue in International Journal of Computer Applications on Advanced Computing and Communication Technologies for HPC Applications, pp. 31-35.

[21] Jain, A.; Kumar, R. (2017): Hybrid load balancing approach for cloud environment. International Journal of Communication Networks and Distributed Systems, 18(3-4), pp.264-286.

[22] Wickremasinghe B.; Calheiros R. N.; Buyya R. (2010): Cloud analyst: A Cloudsim-Based Visual Modeler for Analyzing Cloud Computing Environments and Applications, In the Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 446-452.