IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-12, Dec 2018 E-ISSN: 2347-2693

An Optimal Load Balancing Strategy for Virtual Machines in Cloud Environment

Megha Dubey[1*], Pradeep Singh Chauhan[2]

[1]Dept. of Computer Science and Engineering, OP Jindal University, Raigarh (C.G.), India
[2]Dept. of Electrical and Electronics Engineering, OP Jindal University, Raigarh (C.G.), India

[*]*Corresponding Author: megha.cse617@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 07/Dec/2018, Published: 31/Dec/2018

Abstract— Cloud computing is rising as a leading edge that delivers secure data storage center, unlimited computational resources, network connection and flexible data processing capabilities. The main aim of cloud data centers is how to distribute and maintain the tasks coming from request group efficiently and accurately. Load balancing is one of the major issues in cloud computing which stabilizes the dynamic workload evenly spread across all the virtual machines (VMs) of the cloud data centers. It helps to upgrade the performance of the cloud system and make proper utilization of resources by legitimate allocation. This paper focuses on developing hybrid load balancing algorithm which disseminates the arriving requests in cloud data centers. The proposed algorithm will be implemented using Cloud Analyst simulator and the performance of this algorithm will compare with Throttled, Round Robin and ESCS (equally spread current execution) algorithms on the basis of overall response time and data center processing time. The analysis carried out in the paper shows that the proposed algorithm performs better than the existing algorithms.

Keywords—Load balancing, Existing Algorithms, Hybrid Load Balancing Algorithm, Cloud Analyst

I. INTRODUCTION

Cloud computing is a web-based approach where computing has been pre-installed and exist as a service. All the applications and files hosted on a cloud which consists of thousands of computers interlinked together in a complex way [2]. The main idea behind cloud computing is to provide the ability to store, share and access the data, resources, and services over the internet without any hardware need [7]. Cloud computing is the bundle of virtual resources and services through the internet. As the number of users growing regularly in a cloud platform, load balancing has become major issues for cloud service provider. The goal of load balancing is to utilize the cloud resource in such a way that throughput, efficiency and response time type of measure should be increased. Load balancing balances the load evenly across all the nodes and executes the task in the shortest period of time [14].

In this paper hybrid approach used for load balancing using Throttled and Equally Spread Current execution (ESCE) algorithms. Efficient and enhanced Hybrid scheduling algorithm that can maintain the load and provides modified resource allocation techniques. Hybrid Load Balancer algorithm used to balance the load coming from different data sizes requests groups among virtual machines in cloud data center using virtual machine state list and current allocation of VMs. Results shown that the proposed algorithm can achieve better overall response time and data processing time as compared to earlier load balancing algorithms.

II. LOAD BALANCING

Load balancing is a method that balances the excess dynamic workload evenly across all the cloud data centers. Load balancing is used for attaining better service provisioning, resource utilization and improving the overall performance of the system in terms of throughput, efficiency, fault tolerance and response time. For the proper load distribution, a load balancer is used to decide which virtual machine should assign to which request group [7]. In the cloud, load balancing is done by the process called virtual machine migration [1]. When one data center fails in the cloud, the cloud services can be transferred to another data centers. Services are therefore nonstop [13]. After allocating the task to the virtual machine, cloud task scheduler start to perform load balancing operation so that task can be transfer from overloaded virtual machine to under loaded virtual machine and all virtual machine should stay in balance condition.

Load balancing algorithms can be broadly classified into static and dynamic. Static Load Balancing algorithm does not depend on the recent status of the system and the current state of data. It uses preconception and earlier knowledge about the task, requesting groups and the obtain resources. Unlike Static algorithm Dynamic Load Balancing algorithm does not need previous knowledge about the system. In this algorithm, the workloads are allocating in run time according to the current status and availability of the resources. These algorithms are considered as complex but have better fault tolerance and overall performance over static algorithms [14].

As there are many Load Balancing algorithms in cloud infrastructure which can be used to improve the overall performance by evenly workload distribution. In Round Robin algorithm, the assigned jobs to each resource according to time slices (also known as time quantum) in circular order without any priority consideration. This algorithm is a type of static algorithm which uses the round robin fashion for allocating job. It selects the first task randomly then equally distributes all those tasks to multiple nodes according to their time slices. Because of the different processing time of each job when it will assign to resources some nodes get overloaded and some nodes get under loaded at many points in time [11].

Active Monitoring algorithm balances the tasks among available virtual machines to even out the number of active tasks on each virtual machine at any given time. In this algorithm active load balancers maintain all the details about the VM whenever any request comes to data center controller it will parses the table for available VM in active load balancer. Active load balancer returns the VM Id then data center controller distributes equal amount of workloads on all the available virtual machines [8]. In Throttled algorithm data center controller explores a available virtual machine for allocating the specific request. In this algorithm throttled load balancer maintains a list of Busy/Available VMs. Whenever a request arrives it parses the list of VM when the first available VM found the load balancer returns the VM Id and ensures that only a pre-defined number of request groups are assigned to a single VM. If more request groups are present than the number of available VMs, then the requests groups will have to be queued until the data center controller finds next available VM [10] [8].

III. RELATED WORK

In [3] Tripathi et.al. proposed the hybrid algorithm of ant colony optimization and bee colony algorithm. This mechanism takes the characteristic of a complex network into consideration. The bee colony algorithm is combined with the Ant colony optimization algorithm and the results obtained are better than the individual ACO algorithm. Domanal et.al [4] has discussed a hybrid scheduling approach for load balancing in a distributed cloud environment by combining the two algorithm Divide-and-Conquer and Throttled algorithms referred to as DCBT, which reduces the total execution time of the requests and thereby enhance the average resource utilization. R Somani et.al. [5] Discussed the hybrid methodology for VM level load balancing using concepts of two traditional algorithms for load balancing are Round Robin Algorithm and Throttled algorithm. The proposed hybrid algorithm was found to be efficient in case of same data size per request as well as for different data size per request. S. Mohapatra et.al. [6] Implemented four algorithms named as RR, Throttled, ESCE, FCFS for the balance of load using cloud analyst tool. They also give a comparison of various policies utilized for load balancing and the performance of the Round robin algorithm proved the best among all other.

IV. METHODOLOGY

The goal of the proposed work is to design an efficient scheduling algorithm that uniformly distributes the workload among the available virtual machines in a data center and at the same time, reduce the overall response time and data center processing time. The proposed approach incorporate Throttled and ESCE algorithm Fig.1. In throttled algorithm virtual machines states are used, they are either AVAILABLE or BUSY. AVAILABLE state indicates that the virtual machine is accessible for cloudlets allotment, where BUSY state indicates that the current virtual machine is occupied by earlier cloudlets and is unavailable to handle any new cloudlets request. This current load state of a VM helps in taking a decision whether to allocate cloudlets to virtual machines or not. Active monitoring Load Balancing algorithm repeatedly monitors the list of cloudlets currently allocated to each virtual machine. This allocated cloudlets list helps in determining whether a VM is overloaded or under loaded. On the basis of currently allocated VM, Hybrid load Balancer moves some load from overloaded VMs to the under loaded VMs. After the balanced allocation the task transfer time or data center processing time and overall response time will be calculated. The Result shows that hybrid algorithm performs better than all existing algorithm.

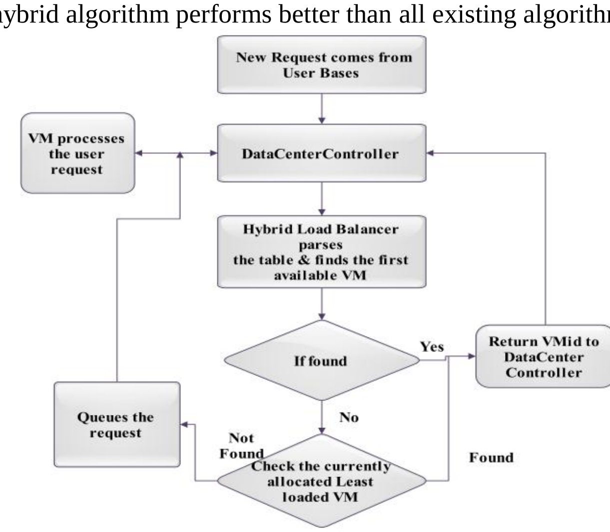

Fig.1 Flow Chart of Hybrid Load Balancer

A. Hybrid Load Balancing Algorithm

- Step1. User bases UB1, UB2...UBn Available VMs VM1, VM2, VM3….VMn within data center. Initially all VMs are available.
- Step2. Hybrid Load Balancer maintains an index table of VMs with the state of the VM (BUSY/AVAILABLE) and the number of requests currently allocated to the VM.
- Step3. Data Center Controller (DCC) receives a new request. DCC queries the Load Balancer for the next allocation.
- Step4. If state of VM is equals to AVAILABLE, Then return the VM Id.
  Go to step 6
- Step5. Else Check the current allocation count of least loaded VM if found return the VM Id.
- Step6. The DCC sends the request to the VM identified by that id and notifies the Load Balancer about new allocation.
- Step7. If not found Load Balancer returns -1, and The DCC queues the request.
- Step8. When the VM finish processing the request and the DCC receives the cloudlet response; it notices the load balancer for VM de-allocation.
- Step9. The load balancer updates the status of VM in VMs state list and index table.
  - Step10. Continue from step 3.

B. Experimental Setup

In this paper, the proposed Hybrid Load Balancing algorithm is implemented using the following software and tools: windows10 operating system, Eclipse Neon IDE, JDK 1.8 and cloud analyst tool. This algorithm is implemented for IAAS (Infrastructures as a Service) model in a simulated cloud environment. Cloud Analyst is an extension to cloud sim oriented GUI based tool used for modeling and analyzing of large cloud environment [8]. Some key elements of Cloud Analyst simulator are as follow:

- User Base - A User Base models a group of users that is considered as a single unit in the simulation. For the implementation purpose we defined user bases in 6 regions globally with the following parameters in Table1 [11].
- Datacenter Controller - Data Center object and manages the data center management activities such as VM creation and destruction and does the routing of user requests received from User Bases via the Internet to the VMs [12].
- Internet Cloudlet - An Internet Cloudlet is a grouping of user requests [9].
- VM Load Balancer - This component models the load balance policies that are used by data centers when serving allocation requests of users.
- Simulation Duration - 60 min

Table 1 Parameters used for user bases configuration

Name	Region	Request per User per hr	Data Size per request (bytes)	Peak hours start (GMT)	Peak hours Ends (GMT)	Avg Peak Users	Avg off Peak Users
UB1	0	80	100	8	10	500000	90000
UB2	1	80	200	14	16	500000	50000
UB3	2	80	300	16	18	400000	40000
UB4	3	80	400	18	20	200000	20000
UB5	4	80	150	20	21	180000	18000
UB6	5	80	50	21	23	300000	30000

- Cloud App Service Broker - This component models the service broker policies that handle traffic between user bases and data centers [8].
- Service Broker Policy - Optimize Response time Policy and Closest Data Center policy.

Table 2 Parameters used for data Center Configuration

Parameter	Value Used
VM Image Size	10000
VM Memory	1024 Mb
VM Bandwidth	1000
Data Center – Architecture	X86
Data Center – VMM	Xen
Data Center – OS	Linux
Data Center – Number of Machines	20
Data Center – Storage per machine	100000 Mb
Data Center – Number of processors per machine	4
Data Center – Memory per Machine	2048 Mb
Data Center – Available BW per Machine	10000
Data Center – Processor speed	100 MIPS
Request Grouping Factor	100
User Grouping Factor	1000
Executable Instruction Length	250
Data Center – VM Policy	Time Shared

V. RESULTS AND DISCUSSION

In this paper, the comparative analysis of three prior load balancing algorithm Round robin, ESCS Throttled with the Hybrid algorithm is developed using Cloud Analyst. The various load balancing policies are used to handle the traffic load between the user bases and virtual machines of data centers. This paper focuses on minimizing the average overall response time and average data center processing time of user bases and data centers. We have reconfiguring cloud analyst setup by adding the number of user bases from UB1…UB6 and the data size of requests taken constantly.

Table 3:- Simulation result of closest DC policy

Closest Data Center Policy		
Algorithm	**Overall Response Time**	**DC Processing Time**
No of Virtual Machines 5		
ESCS	438.95	158.10
THROTTLED	358.51	79.31
HYBRID	358.43	79.31
RR	438.97	158.04
No of Virtual Machines 10		
ESCS	439.14	158.29
THROTTLED	358.55	79.38
HYBRID	358.45	79.33
RR	438.96	158.11
No of Virtual Machines 15		
ESCS	412.42	131.85
THROTTLED	346.41	67.32
HYBRID	346.35	67.31
RR	410.50	129.93

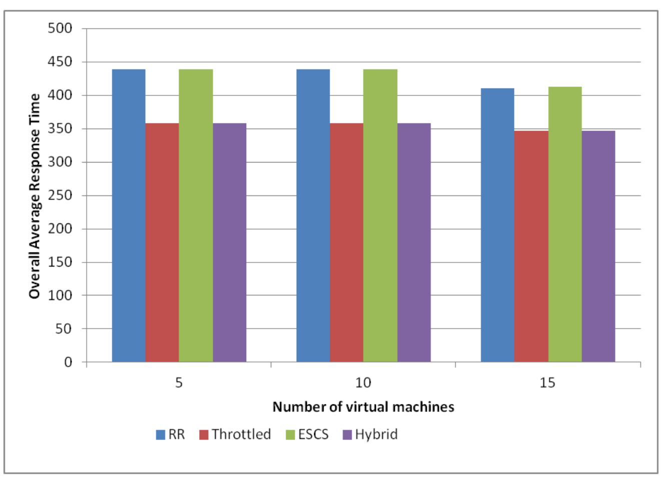

Fig.2 Performance analysis of Overall Response Time in Closest DC policy

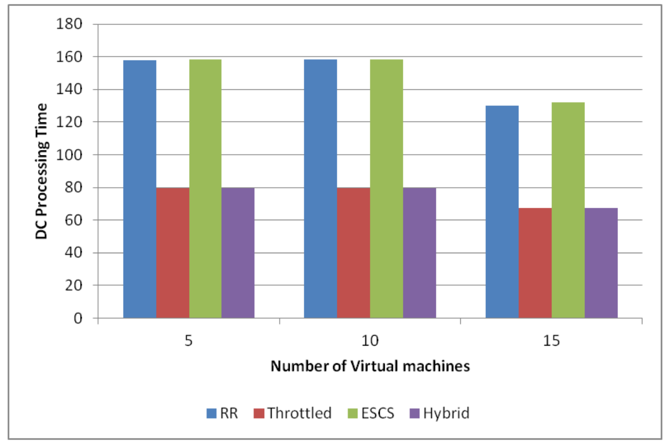

Fig 3 Performance analysis of DC Processing Time in Closest DC Policy

The numbers of data centers are considering in the different region and we have changed the value of virtual machines from 5 to 15. In every iteration, we have increased the values of virtual machines from 5-15 in each data center and run the simulation. According to table3,4 we have calculated different values of overall response time and data center processing time for different numbers of virtual machines. The outcome shows clearly in graph that the proposed hybrid load balancer performs better than existing algorithms. Fig 5-6 shows the comparison between different load balancing algorithms on the basis of overall response time and data center processing time in different values of virtual machines.

Table 4: Simulation result of optimized response time policy

Optimized Response Time Policy		
Algorithm	**Overall Response Time**	**DC Processing Time**
No of Virtual Machines 5		
ESCS	407.89	69.89
THROTTLED	373.03	34.74
HYBRID	372.51	34.18
RR	406.35	68.31
No of Virtual Machines 10		
ESCS	383.52	45.24
THROTTLED	373.47	35.17
HYBRID	362.40	23.61
RR	380.08	47.71
No of Virtual Machines 15		
ESCS	381.27	42.99
THROTTLED	362.36	23.57
HYBRID	361.95	23.13
RR	380.06	41.70

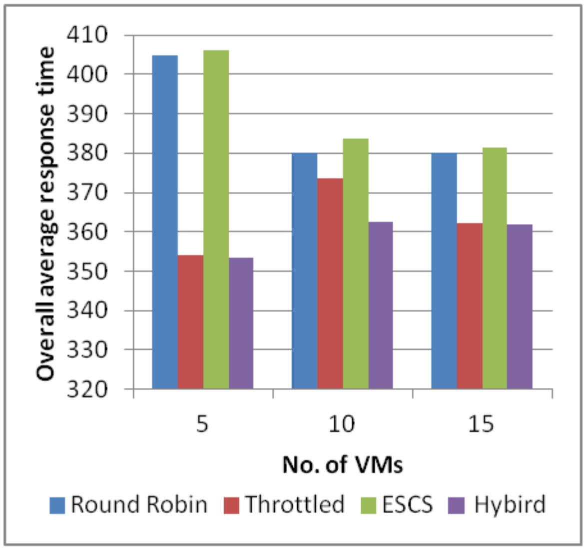

Fig. 4 Performance analysis of Average Response Time in Optimized Response Time Policy

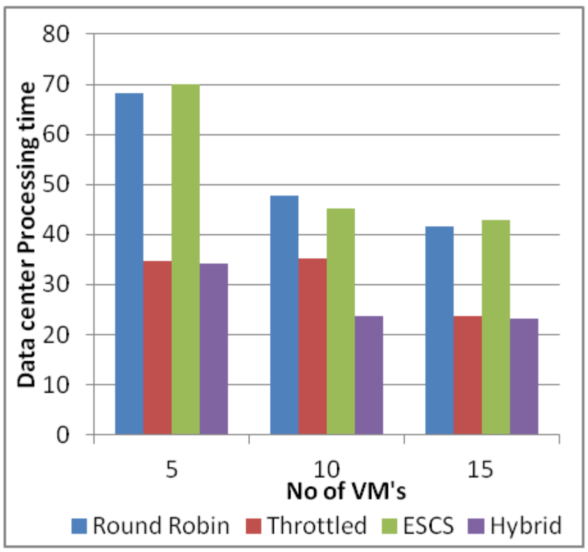

Fig.5 Performance analysis of Data Center Processing Time in Optimized Response time Policy

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we have proposed the Hybrid load balancing algorithm which is utilizing the features of ESCS and throttled load balancer algorithm. The vital part of this paper is designing hybrid algorithm and comparison with pre existing load balancing algorithms based on different types of parameters like response time and data center processing time. We also described the performance analysis graph which showing the results in different conditions. Our algorithm minimizes the value of average response time and data center processing time as compared to existing algorithm.

REFERENCES

[1] Zhen Xiao, Weijia Song and Qi Chen, "*Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment*", IEEE transactions on parallel and distributed systems , vol. 24, no. 6, 2013.

[2] Dhinesh Babu L.D., P. Venkata Krishna, "*Honey bee behavior inspired load balancing of tasks in cloud computing environments*", Applied Soft Computing Science Direct, 2292–2303, 2013.

[3] Apoorva Tripathi, Saurabh Shukla and Deepak Arora, "*A Hybrid Optimization Approach for Load Balancing in Cloud Computing*", Advances in Computer and Computational Sciences, Advances in Intelligent Systems and Computing, Springer,554, 2018.

[4] Shridhar G.Domanal and G. Ram Mohana Reddy,*"Load Balancing in Cloud Environment using a Novel Hybrid Scheduling Algorithm",* IEEE International Conference on Cloud Computing in Emerging Markets, 2015.

[5] Rajkumar Somani, Jyotsana Ojha, *"A Hybrid Approach for VM Load Balancing in Cloud Using CloudSim, International Journal of Science",* Engineering and Technology Research (IJSETR), vol 3, Issue 6, 2014.

[6] S. Mohapatra, K. S. Rekha and S. Mohanty, *"A comparison of four popular heuristics for Load balancing of Virtual Machines in Cloud Computing",* International journal of Computer Applications, vol 68, no 6, pp–38, 2013.

[7] Sukrati Jain, Dr. Ashendra K. Saxena , *"A Survey of Load Balancing Challenges in Cloud Environment",* IEEE 5th International Conference on System Modeling Advancement in Research Trends, 2016.

[8] B. Wickremasinghe, *"CloudAnalyst: A CloudSim-based Tool for Modeling and Analysis of Large Scale Cloud Computing Environments",* MEDC Distributed Computing Project, CSSE Dept., University Of Melbourne,2009.

[9] R. Buyya, R. Ranjan, and R. N. Calheiros, *"Modeling And Simulation Of Scalable Cloud Computing Environments And The Cloudsim Toolkit: Challenges And Opportunities",* Proc. Of The 7th High Performance Computing And Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.

[10] Meeta Singh, Poonam Nandal, and Deepa Bura, *"Comparative Analysis of Different Load Balancing Algorithm Using Cloud Analyst",* Springer Nature Singapore Pte Ltd. B. Panda et al. (Eds.): REDSET 2017, CCIS 799, pp. 321–329, 2018.

[11] Bhathiya Wickremasinghe ,Roderigo N. Calherios, *"Cloud Analyst: A Cloud-Sim-Based Visual Modeler For Analyzing Cloud Computing Environments And Applications",* Proc Of IEEE International Conference On Advance Information Networking And Applications,2010.

[12] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, *"Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility",* Future Generation Computer Systems Elsevier Science Direct, , vol 25, no 6, pg. 599–616, ISSN: 0167-739X, ,Amsterdam, The Netherlands June 2009.

[13] Shang-Liang Chen , Yun-Yao Chen , Suang-Hong Kuo, *"CLB: A novel load balancing architecture and algorithm for cloud services",* Computersand Electrical Engineering, Elsevier Science Direct, 2016.

[14] Sushil Kumar, Deepak Singh Rana, *"Various Dynamic Load Balancing Algorithms in Cloud Environment: A Survey",* International Journal of Computer Applications, 0975–8887 Vol 129 No 6 2015.

