International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 05 - Issue 06 || June 2019 || PP. 55-60

Hybrid Load Balancing Algorithm Implementation in cloud computing environment

Dina Farouk Altayeb[1], Fatima Abdelghani Mustafa[2], prof. Amin bebiker[3], Dr. Ashraf Gasim Elsid[4]

[1]*Future University, Faculty of Engineering, Department of Computer Engineering)*
[2]*Future University, Faculty of Engineering, Department of Computer Engineering)*
[3]*Neelain university, Faculty of Telecommunication Engineering Department*
[4]*Future University, Dean of Faculty of Telecommunication Engineering*

Abstract: In Cloud computing day by day number of users are increasing with large amount of data processing requirements. The biggest challenge for cloud data centers is how to service millions of requests arriving very frequently from end users efficiently and correctly. Therefore, Load balancing is the optimal approach solving the issue of growing demands for the resources of data centers. The need to efficient and powerful load balancing algorithms is one of the most important issues in cloud computing to improve the performance. This paper proposed hybrid load balancing algorithm to improve the performance and efficiency in cloud computing environment. The algorithm considers the advantages of random, greedy, and Throttled algorithms. The hybrid algorithm has been evaluated and compared with other algorithms using cloud Analyst simulator in hardware complexity environment. The experiment results show that the proposed algorithm improves the average response time and average processing time compared with other algorithms. This approach is used to minimize the response time, avoid the bottleneck problem, maximize the services and reduce the machine cost.

Keywords: Cloud computing, Load Balancing, Data Center, Virtual Machines, Cloud Analyst

1. Introduction:

In recent years, Cloud computing become a new computing model emerged from the rapidly development of internet. It leads the new IT revolution. Cloud computing considered an evolution of distributed systems. The National Institute of Standards and Technology's (NIST) define a Cloud computing as "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. "[3] cloud computing has moved computing and data away from desktop and portable PCs into large data centers [2]. Cloud computing is a heterogeneous environment offers a rapidly and on-demand wide range of services [1]. Heterogeneous environment means having different hardware characteristics including CPU, memory, storage and other hardware [2]. Load balancing considered as one of the most challenges in cloud computing. It is the major factor to improve the performance of the cloud computing. The current load balance scheduling algorithms in cloud computing environment is not highly efficient [5]. Load balancing in cloud computing environment is very complex task till today, because prediction of user request arrivals on the server is not possible, and each virtual machine has different specification, so it becomes a very difficult to schedule job and balance the load among node [6].

Cloud Computing can be considered as a platform for development, maintenance and accessing applications by user by paying for the resources which are only used for certain time. To serve for huge number of requests from different types of users located at different parts of the world on a pay-per-usage bases, the process of virtualization has been followed in cloud computing environment. There are so many service providers who are responsible for maintaining the application on cloud environment. The most predominant cloud service providers are Google, Amazon, Microsoft and many others. In order to serve the huge traffic to world these service providers maintain data centers all over the world where the data is stored in bulk and requests are processed [4]. Depending on the number of requests to be processed by a data center, no. of virtual machines is created where on a single CPU different operating system and their configurations can be run. It gives the illusion to the users that, there are numbers of CPUs are involved for processing the requests. This whole process can be considered as virtualization and it is monitored by the software called as hypervisor. The hypervisor is mainly responsible in creating and maintaining the virtual machines. The requests from all the users are considered as individual request and each of the virtual machines is assigned a request and effort is made to keep them busy for longer time. This is considered to be load balancing policy which makes an effort to maximize the throughput of virtual machines. Job Scheduling is a process of allocating jobs onto available

resources in time. It is also defined as the process of finding an efficient mapping of tasks to the suitable resources so that the execution can be completed with the satisfaction of some objective functions [6]. The objective functions could be such as minimization of execution time as specified by customers and maximization of resource utilization as specified by service providers. Efficiency of scheduling algorithm directly affects the performance of the system with respect to delivered Quality of Service. In short, more efficient is the scheduling algorithm, better is the Quality of Service delivered [7].

Many research works have proposed a load balancing algorithm in cloud computing such as Round Robin, Equally Spread Current Execution and Throttled Load Balancing Algorithm. The current load balance scheduling algorithms in heterogeneous cloud computing environment is not highly efficient [5]. This research proposes a hybrid load balancing algorithm to improve the performance and efficiency in heterogeneous cloud computing environment. The proposed algorithm takes advantages of random, Throttled and greedy algorithms and considers the current resource information and the CPU capacity factor to achieve the objectives. The hybrid algorithm has been evaluated and compared with other algorithms using cloud Analyst simulator. The result showed improvements on average response time and on processing time by considering the current resource information and the CPU capacity factor compared with other algorithms, and this means the performance has improved. The paper in general will be organized as follows: Section two is devoted to cloud computing overview. In section three the load balancing overview. In section four the related works. In section five we define the proposed algorithm. Section six is about experiment and results [8].

2. Load Balacing:

Load balancing is defining 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 [9].

3. Related Work:

Load balancing is used to distributing the load across multiple nodes for enhancing the overall performance of a system. The current load balancing algorithms in cloud computing environment is not highly efficient. Load balancing is very complex task today, because the users request arrival on server is not possible i.e., we cannot predict it. Each Vms [1] [4] has different specifications. So, it is difficult to schedule the job and balance the nodes. Recently, many research works done on load balancing. Load balancing mainly classified into two categories, static load balancing and dynamic load balancing algorithms. Static load balancing algorithms mainly defined in the design or implementation of the system. Dynamic load balancing algorithm considered only current state of the system during load balancing. The existing algorithms are following:

Jinhua Hu; Jianhua Gu [10] proposed a new algorithm to enhance job scheduling using genetic algorithm. The algorithm used a historical data and current state of the system. And it makes a mapping relationship between the set of physical machines and the set of VMs. It chooses the least-affective solution by computing ahead influence of the system after the deployment of the needed VM resources. They used some equation to find the best scheduling solution using population. The results showed an improvement in the utilization of resources. On the other hand, the proposed algorithm has high cost to store and retrieve the historical data of the system nodes, and this may also increase the response time and the processing time in a heterogeneous environment consist of heterogeneous resource.

Martin Randles, David Lamb, A. Taleb-Bendiab [11] proposed an ant colony algorithm as a solution for load balancing in the cloud. Ants depend on the strength of pheromone to select the optimal path that leads to their destination. In the same way each node in the network has a pheromone. Each row in the pheromone table represents the routing preference for each destination, and each column represents the probability of choosing a neighbor as the next hop. If an ant is at a choice point when there is no pheromone, it makes a random decision. If the pheromone is existing, the node with high probability is selected and then the pheromone table is updated

by increasing the probability of this node and decreasing other nodes probabilities. The main drawback of this algorithm is that it does not consider the fault tolerance issues.

Singh, Bedi and Gupta in [12] develop a new heterogeneous load balancing algorithm to distribute the load across number of servers, they create VM's of different data center according of host specification including core processor, processing speed, memory, storage etc. Then allocate weighted count according to the RAM allocated to the VM's in the datacenter. Then used a data structure to maintain weight count and the current allocation count of the VM, they allocate the VM which have available status and have a higher RAM. When allocates a new VM, the algorithm returns the VM id to the Data Center Controller, then updates the allocation count for that VM and adding the new allocation to the busy list. When the VM finishes processing the request the algorithm de-allocate the VM and removed the VM from the busy list. The main drawback of the algorithms was the authors did not present any results and any comparison with other algorithms.

4. Problem Identification:

Cloud computing is a term, which involves virtualization, distributed computing, networking, software and web services". As we talk about a cloud it consists many parameters like shoppers, datacenter & distributed system. Cloud comprises of fault tolerance, convenience, and quantifiability, litheness, compact overhead for users, compact value of possession etc. [2].

Load balancing is therefore may be defined as the method of allocate the load among different nodes of a Data Center to enhance each resource employment and process latency whereas additionally avoiding a state of affairs wherever a number of nodes are highly loaded whereas alternative nodes are idle or doing little or no work, and some of the physical machines and virtual machines are having maximum imbalance level of Cloud data centers [5].

5. Solution Domain:

Enhanced hybrid approach is the advancement of hybrid algorithm which contains combination of randomizing, greedy Throttled load balancing algorithm. Enhanced Hybrid algorithm maintains an index list of VM allocation status as well as list to count the allocated request. The allocated request list is compared with the VMs index list. If VMs indexlist is greater than allocated request list it means that VMs are available to take request else request has been queued until VM is been available. If the VM has been queued, it has to wait in the queue itself. So new host has been created using host create function. In case of availability of VM, the jobs are allocated to that particular VM. And both the index list and hash list are updated. The job in queue need not wait for long time for the virtual machine to become available. Maximize the resource utilization than the existing system. Minimize the response time and negligible idle time.

6. Proposed Algorithm:

Step 1	Initialize the Hybrid VM load balancer, it maintains an index of VMs and state of the VMs (busy/available). At start all VMs have zero allocation.
Step 2	Sort all available VM in VM-queue in Ascending form.
Step 3	A new request has been sent to the Data Center Controller (DCC).
Step 4	The Datacenter Controller receives the user requests/cloudlets
Step 5:	Send Requests to the Least VM load in VM-Queue
Step 6	Check the current available VM load is overload or not If yes: distribute some requests to the next least VM load in VM-Queue If no: go to step 7
Step 7	Execution tasks
Step 8	Update VM-Queue by recalculate VM load (busy/available)
Step 9	Check all user requests are executed If yes: go to step 10 If no: go to step 3
Step 10	End

The load balancer spreads the load on to completely different nodes with available VM (have least load). The load balancer maintains a queue of the roles that require to use and are presently mistreatment the

services of the virtual machine. The balancer then unendingly scans this queue and therefore the list of virtual machines.

We implemented hybrid algorithm, on Eclipse using advance java, and cloud Simulation CloudAnalyst. we configure many parameters like number of datacenters, number of cloudlets, VM configuration, bandwidth and MIPS. We implemented Four algorithm of load balancing are:

- Throttled.
- Equally Spread Current Execution.
- Round Robin.
- Hybrid Algorithm.

7. Implementation and Results:

The initial process in this Enhanced hybrid algorithm is to configure the user base such that name, region request per user per hour, data size per request, peak hours start, average peak users and average off-peak hours. Moreover, the duration of the simulation is fixed your requirement and the service broker policy (optimal response time) is also selected. These are all done in the main configuration in cloudAnalyst. Using data center configuration in cloud analyst we add few data centers with its region, architecture, OS, virtual machine manager, cost per VM, memory cost, storage cost, data transfer cost and physical hardware units. The advanced configuration such as user grouping factor in user bases, request grouping factor in data centers and executable instruction length per request are done. The proposed algorithm is selected as the load balancing policy across VM's in a single data center.

Cloud load balancing is developed in this research with help Java (JDK 8.201) and Eclipse 12.4.10 on window operating system 10. In Result Analysis Compare Proposed system with existing system in term of Average Response time and CPU Factor.

First step: This is the first page of our project, which is shown in figure1.

In this step we use 4 data center on different region (US, UK, AUSTRILLIA, AND AFRICA), 50 VM on each data center, and user 25 user

Figure 1: Cloud Analyst Simulator front page First Step

Second Step: Adding Hybrid Algorithm in CloudAnalyst.

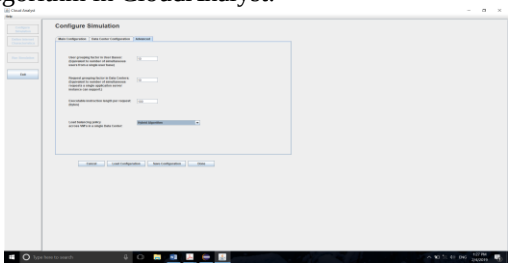

Figure 2: Adding Hybrid Algorithm in CloudAnalyst.

Third Step: Configure User base, and four Datacenter and Implement VMs with different CPU factors.

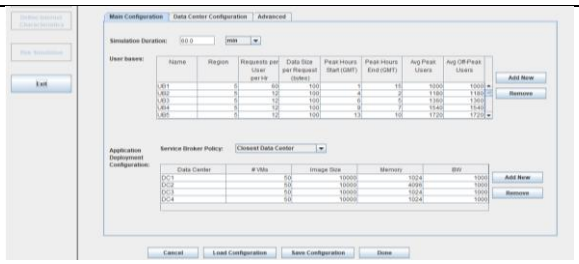

Figure 3 Configure Datacenter and Implement VMs.

Fourth Step: run simulation for all Algorithm

Figure 4: Demonstration of proposed work in Third step.

Fifth Step: we repeat all step above for all other load balancing algorithm and round robin, ESCE, Throttled

Sixth Step: compare the results with other in term of Average time

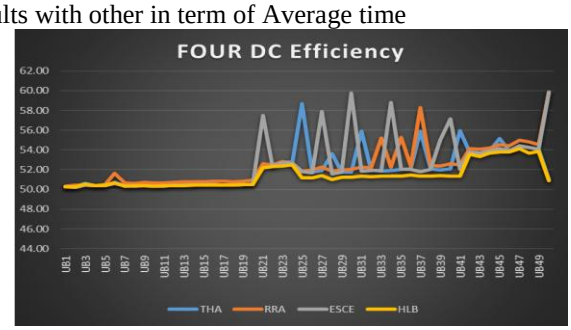

Figure 5 Four Data Center Efficiency on Different Region (US-UK, AUSTRILLIA, AND AFRICA)

Table 1: shows average, Minimum, and Maximum time for each Load Balancing algorithm used in this paper.

	Hybrid Algorithm	**Round Robin**	**Throttled**	**ESCE**
DC RT (Avg)	54.0	56.73	56.31	57.18
DC RT(Min)	38.25	39.75	39.37	39.78
DC RT(Max)	169.55	232.83	169.48	440.15

The proposed hybrid algorithm was efficient better from all other load balancing algorithm in case of same data size per request as well as for different data size per request as shown on figure below figure 6:

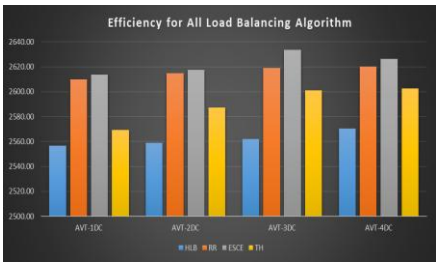

Figure 6. Efficiency for all load balancing Algorithm

8. Conclusion and Future Work:

The hybrid approach load balancing algorithm is proposed and implemented in cloud computing environment using CloudAnalyst toolkit, in java language. This work involves the combination of Throttled, Random, and Greedy Algorithms. The advancement of hybrid load balancing algorithms is Intermediate deliverables included studying the existing VM load balancing algorithms, proposing an efficient algorithm for VM load balancing, implementing the algorithm on Cloud Analyst, and comparing the proposed algorithm with the existing algorithms on identified parameters with a complex hardware. By analyzing the parameters in graphs and tables we came to know that the overall response time, data processing time is relatively minimized as well as data transfer cost is reduced.

The future work can combine other load balancing approaches that can be applied to balance the load in clouds computing environment, will improve performance of cloud load balancing. In Future also compare load-balancing algorithms on other parameters.

References:

[1]. Velagapudi Sreenivas, Prathap.M, Mohammed Kemae, "Load Balancing Techniques: Major Challenge in Cloud Computing –A Systematic Review". Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pages 445l.

[2]. VeerawaliBehal, Anil Kumar, "Cloud Computing: Performance Analysis of Load Balancing Algorithms in Cloud Heterogeneous Environment", IEEE 2014.

[3]. Klaithem Al Nuaimi1, Nader Mohamed1, Mariam Al Nuaimi1, and Jameela Al-Jaroodi2, "A Novel Approach forDual-Direction Load Balancing and Storage Optimization in Cloud Services", 2014 IEEE 13th International Symposium on Network Computing and Applications.

[4]. Raza Abbas Haidri, C. P. Katti, P. C. Saxena, " A Load Balancing Strategy for Cloud Computing Environment", International Conference on Signal Propagation and Computer Technology (ICSPCT), 2014.

[5]. VasudhaArora,S.S.Tyagi, "Performance Evaluation of Load Balancing Policies across Virtual Machines in a Data Center", 2014 International Conference on Reliability, Optimization and Information Technology -ICROIT 2014.

[6]. Nagamani H.Shahapure, Jayarekha P, "Load Balancing With Optimal Cost Scheduling Algorithm", International Conference On Computation of Power, Energy, Information and Communication (ICCPEIC), 2014.

[7]. Shaoyi Song, TingjieLv, Xia Chen, "A Cooperative Game Method for LoadBalancing inCloud Based on Cost-Efficiency", ICUFN 2014.

[8]. Tingting Wang, ZhaobinLiu. , Yi Chen, YujieXu, "Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing", 12th International Conference on Dependable, Autonomic and Secure Computing, IEEE 2014.

[9]. Vishwas Bagwaiya1, Sandeep k. Raghuwanshi, "Hybrid Approach Using Throttled and ESCE Load Balancing Algorithmin Cloud Computing", International Conference,2014.

[10]. Jinhua Hu; Jianhua Gu; Guofei Sun; Tianhai Zhao; , "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment" Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on, vol., no., pp.89-96, 18-20 Dec. 2010.

[11]. Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[12]. Singh, A., R. Bedi, and S. Gupta, Design and implementation of an Efficient Scheduling algorithm for load balancing in Cloud Computing. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2014. 3(1).