IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol-6, Special Issue-3, Apr 2018 E-ISSN: 2347-2693

AMRRHC: Active Monitoring Round Robin with Holding Capacity Load Balancing Algorithm

Bhagyalakshmi[1*], D.Malhotra[2]

[1*]Dept. of CS&IT, School of Applied Sciences, Central University of Jammu, Jammu, India
[2] Dept. of CS&IT, School of Applied Sciences, Central University of Jammu, Jammu, India

[*]*Corresponding Author: bhagyalakshmi.magotra@gmail.com, Tel.: +91-9419272731*

Available online at: www.ijcseonline.org

***Abstract*—** The cloud platforms are becoming popular day by day with the advent of more customers every second, flooding the cloud environment with millions of requests thereby making the processing of such requests a major challenge to be handled. The major goal of cloud computing is to provide requested resources efficiently and effectively which can be achieved by distributing the load in a balanced manner among various nodes leaving the network in an optimal condition. In this research paper, much optimized load balancing scheme has been proposed to schedule the tasks in the cloud environment. The scheme has been designed to calculate the load on the list of available Virtual Machines (VM), considering the CPU utilization and usage as a metric for calculation of load. The proposed scheme modifies the already existing Active Monitoring Load balancing algorithm and merges the advantages of Active Monitoring Load Balancing and dynamic Round Robin scheduling techniques.

Keywords— Cloud computing; resource management; load balancing; virtual machines.

I. INTRODUCTION

Cloud computing has proved to be a boon in the field of computing technology. The cloud computing environment has two main players: the providers and the users, and system comprises of several servers, virtual machines, data centers, storage devices etc which are interconnected in an efficient way. The users request for various resources that they require and the cloud providers grant these resources to the needed users according to their needs. These resources can be released back to the cloud provider after their use. Thus cloud computing provides a multi tenant environment where the resources are used by one user are shared by another. Cloud service providers majorly provide three basic types of services which include software, platform and infrastructure and are termed as SaaS (software as a service), PaaS (platform as a service), and IaaS (infrastructure as a service). Rapid elasticity, demand computing, resource pooling broad network access and measured services are the five basic characteristics of a cloud[1]. The number of users requesting for cloud resources are increasingly linearly but the resources available are limited. Thus there arises the need of proper resource management that must take user satisfaction into consideration. Resource management is the process of assigning virtual machines, computing resources, and storage resources to the users on demand[2]. The term resource management is an umbrella term that includes in it resource provisioning, resource scheduling and resource monitoring. Resource provisioning deals with discovery and selection whereas resource monitoring deals with the resource usage. Resource scheduling is the major activity of resource management process that is responsible for the resource allocation and mapping and hence requires efficient algorithms [3], [4]. In cloud computing, the users ask for various resources time to time and thus the availability of the resources needs to be checked using an efficient algorithm which can schedule these resources according to the need and request of the user. Thus having an efficient task scheduling scheme is of utmost importance so that the resources can be utilized optimally. The aim of such scheduling algorithms is to improve the latency and total response time for the jobs being submitted by the user to the cloud while maximizing the overall utilization of the resources and the throughput.

There are various scheduling issues in cloud computing like cost, power, efficiency, QoS aware resource allocation that need to be addressed by varieties of solutions but they cannot provide high performance without proper balancing of load. The workloads must be submitted to the appropriate hosts in order to achieve maximized utilization of resources with the minimized cost. However, the challenges of unpredictable user requests and the complexities of the cloud environment may imbalance the load. Proper load balancing algorithms can help in using the available resources optimally, thereby minimizing the resource consumption[5]. Hence, an

appropriate mapping between resources and user requests is critical which has lead to the proposal of many load balancing algorithms.

The load balancing algorithms are divided into the categories of being static and dynamic. The static algorithms are the simple and easier to implement without any need to handle the changing needs of the user and the dynamic load of the environment whereas the dynamic load balancing algorithms are the ones that deal with the situations where the needs change dynamically with time. Furthermore, the problem of load balancing can be solved by either assigning the tasks to the nodes in such a manner that the load gets evenly distributed or by migrating the virtual machine from one host to another i.e., the migration from an overloaded host to lightly loaded ones.

This research mainly focuses on the first sub problem of load balancing and is divided into many sections. Section II focuses on the work already done in literature related to the problem of evenly assigning the tasks to balance the load. Section III includes the proposed scheme AMRRHC and its pseudo code. The comparison of AMRHRC with existing round robin and active load monitoring technique has been done in Section IV and finally Section V concludes the proposed research work along with its future scope.

II. RELATED WORK

S.Kapoor and C. Dabas have proposed a cluster based load balancing algorithm[1]. In the proposed scheme the VM's are first divided into clusters based on three parameters. These are bandwidth, CPU processing power and memory. The machines are divided into k clusters using K-means clustering, each cluster having similar resources. And a record of the same is maintained at the load balancer. Whenever DCC receives a request, it checks for the cluster which can handle the request. If more than one cluster is found then the cluster found first on the list is selected for the execution of the required task and user request is executed. The status of the VM is also set to BUSY which is made AVAILABLE after the completion of the task.

In a two level scheduling algorithm proposed by Y. W. Qiu et. al .[6] for the cloud computing environment, the execution of the scheduling process takes place in two phases. Initially for the execution of the tasks, scheduler allocates these tasks to VM's with minimum capacity that can accommodate the request. If no such VM exists then the host is requested for the creation of a new machine. If the host is also overloaded then the task request is queued for execution at some later point. This forms level 2 that constitutes the mapping of tasks to the respective VM's. At level 1, hosts take care of addition and deletion of the VM's depending upon the workload. If the machines are overloaded then new machined are created to balance the load. However if the machines remain idle for some predefined period of time, then the machines are deleted. Thus, this phase deals with dynamic addition or deletion of the VM's.

The load balancing algorithm proposed by N. K. Chien, N. H. Son and H. Dac Loc [7] is based upon the estimated finishing times of various jobs being assigned to the VM's. Whenever the Data Centre Controller receives a request for the execution of a task, it checks the status index table and calculates the finishing time of job on each of the VM including the finishing times of the tasks in the respective queues. For the selection of the VM for the submission of the new request, VM with the earliest finishing time is selected. The calculation of the finishing time is done by different formulas proposed by the authors. Simulation results show that the proposed algorithm shows better response and processing time.

U. Thakkar presented "A Novel Approach for Dynamic Selection of Load Balancing Algorithms in Cloud Computing" [8]. The proposal is a combination of mapreduce, LBMM, Honeybee and antcolony schemes depending on task size and the workload. For each task a computation time with each available resource is calculated and the resource for which the minimum computation time has been calculated is chosen to execute the task. Furthermore, depending upon the workload and response time count, being calculated by authors, the task is reschedule either for ant colony or honeybee algorithms.

The Priority Based Load balancing algorithm proposed by U. Thakkar [9] includes two phases. In the first phase of PLBA, the priorities of the VM's are calculated depending upon the energy efficiency of the VM's and the jobs are assigned to the nodes according to these calculated priorities. In the second phase, the VM's are checked for their load. If any machine is over loaded then the task is shifted to another less loaded machine. In contrast if few machines are under loaded, in that case the workload of these machines is combined and shifted over to one machine that can handle the load and the machine from which the load has been removed is switched off leading to an energy efficient environment.

Shridhar G. Domanal and G. Ram Mohana Reddy [10] have proposed an algorithm VM-Assign which is responsible for distribution of all the coming requests to all the available virtual machines in an efficient manner. Every time a machine is needed, the algorithm looks for the least loaded one. The algorithms also takes care that the same machine is not selected again and again leading to under utilization of the remaining machines. The researchers have used CloudSim Simulator for result analysis and have compared the proposed scheme with the existing Active-VM load balance algorithm. The proposed technique resolves the ineffective utilization of the VMs resources when relates result to previous algorithm.

"Modified throttled algorithm" [11] is an optimizing load balancing technique which works by the distribution of coming jobs request consistently between the servers or virtual machines. As proposed, the algorithm maintains an index associated with each machine and is saved whenever a machine is allocated to the user request . When a new request is encountered the machines are checked for their availability starting from this index. The efficiency is evaluated using simulator Cloud Analyst and result is compared with previous algorithm Round Robin and throttled algorithm. Various versions of throttled algorithm have also been compared [12].

An algorithm has been proposed by R. Panwar and B. Mallick [13] which first checks the availability of the machine by maintaining an index for the total number of allocated machines. If total allocations of VMs are less than the available machines then the available machine is allocated to perform the requested task. However, if no such machine is available then the least loaded machine out of all the machines is selected and is assigned for the execution of the task.

S. Elmougy et. al. [14] have proposed a scheduling algorithm that combines the features of Shortest Job First and Round Robin. The researchers divided the queue of jobs waiting for allocation into two sub queues Q1 and Q2. The Q1 queue holds the shorter jobs whereas Q2 holds the longer tasks. At each round, the median is calculated and depending upon this calculated median the next job in the queue is sent to Q1 if the quantum is shorter than the median and to Q2 if the quantum of the task waiting in the queue is longer than the median of all the time quantum's of all the jobs in the waiting queue. Two jobs from shorter queue and one from longer queue is executed are executed so as to reduce the waiting time.

III. AMRRHC (ACTIVE MONITORING AND ROUND ROBIN WITH HOLDING CAPACITY)

A. *Proposed Scheme*

In this paper AMRRHC has been proposed which overcomes the drawbacks of Active Monitoring Load Balancing (AMLB) and Round (RR). This is done by the calculation of current holding capacity of the machine and choosing the time quantum as the mean of remaining time burst respectively. To begin with, the user sends the request to the Data Centre Controller which requests the VM Load Balancer to execute the request. The load balances maintains a dynamic list which holds the current status of various i.e., whether the machine is BUSY or AVAILABLE. On receiving the request, the load balancer at first looks at the VM state index list to find out if any machine is available. If found, the machine id of the particular VM is returned to the DCC otherwise if no machine is free then the least loaded machine out of all the machines is selected. This least loaded machine is then checked if its remaining capacity is sufficient enough to accommodate the new request. Equation (1) is used to evaluate this. If the result of the equation comes to be greater than 1, then it implies that the remaining holding capacity of the machine is greater than the required one and hence its id is returned to the DCC. Otherwise, if no such machine is found then the request is queued with this least loaded machine and executed in the round robin fashion with dynamic quantum using (2).

B. *Psuedo Code for AMHHRC*

The execution of AMHHRC follows the given steps:

1. Initially all the VM's are available.
2. The request received by the Data Centre Controller (DCC) is forwarded to VM load balancer.
3. VM load balancer maintains the list of all virtual machines along with their status i.e, whether the VM is BUSY or AVAILABLE. Initially al the machines are available.
4. The DCC receives the request for next allocation from the client.
5. If (present allocation of VM's < max. limit of VM's)
   find the first available VM
   return VM id
   otherwise
   find the least loaded VM.
   Calculate α for the selected VM using (1).
   If α>1 (VM is capable of executing the request)
   return VMid
   otherwise
   assign least loaded machine and execute the tasks in RR fashion with dynamic quantum λ , using (2)

$$Holding\ Capacity, \alpha = \frac{actual\ remaining\ capacity}{task\ requirement} \quad (1)$$

$$Time\ Quantum, \lambda = \frac{(\Sigma X_{i \epsilon n})}{n} \quad (2)$$

Where "X" is the remaining time of each job and "n" is the total number of jobs in the queue.

C. *Comparative Analysis With Existing Algorithms*

The proposed algorithm overcomes the disadvantages of existing Active Monitoring Load Balancing and Round Robin algorithm. The benefits and shortcomings of AMLB, RR and AMRRHC have been listed in Table 1.

Table 1. Advantages and Disadvantages of AMLB, RR and AMRRHC

Approach	Advantages	Disadvantages
Active Monitoring Load Balancing (AMLB)	Mitigates bottleneck situations. Improves significantly response time.	Actual instant power of VMs and job size are not considered. This may result assigning jobs to improper VMs.
Round Robin (RR)	Since each user request gets the same CPU time slot for execution, all of them are executed with same priority. The problem of starvation doesn't occur because each process gets equal CPU time for execution and hence no process is starved for execution.	If Time quantum (TQ) is large, it will behave like FCFS and if the TQ is small then many context switches will occur leading to more time consumption.
Active Monitoring Round Robin with Holding Capacity (AMRRHC)	Actual instant power of VMs and job requirement are considered. Dynamic TQ for round robin has been used.	Does not consider the remaining execution time of the tasks running on VM's which might lead to inappropriate VM selection.

D. Flowchart for AMRRHC

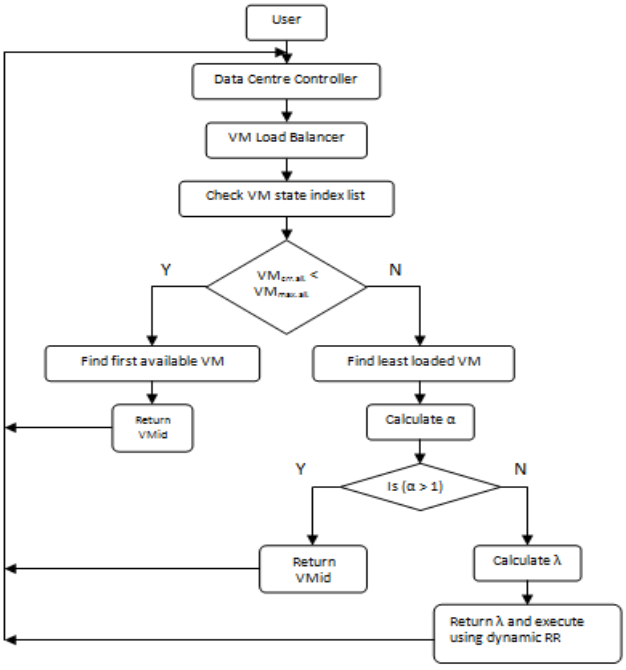

Fig. 1 shows the flow of activities that take place during the execution of the proposed algorithm AMRRHC.

IV. CONCLUSION AND FUTURE SCOPE

Various load balancing techniques have been proposed in literature for reducing the load in the network by actually balancing it. Load balancing increases the efficiency and response time of the network. The work presented in the paper proposes a novel merged load balancing approach taking into consideration the CPU utilization. In future, the same can be considered for dynamic networks and other metrics like bandwidth, memory etc. can also be considered to further optimize the balancing of load in the network and improving the response time and throughput of the network.

REFERENCES

[1] S. Kapoor, "Cluster Based Load Balancing in Cloud Computing," *Ieee*, 2015.

[2] S. Hamid, H. Madni, M. Shafie, A. Latiff, Y. Coulibaly, and M. Abdulhamid, "Recent advancements in resource allocation techniques for cloud computing environment : a systematic review," *Cluster Comput.*, 2016.

[3] S. Singh and I. Chana, "Cloud resource provisioning: survey, status and future research directions," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 1005–1069, 2016.

[4] S. Singh and I. Chana, "A Survey on Resource Scheduling in Cloud Computing :," *J. Grid Comput.*, pp. 217–264, 2016.

[5] M. Xu and W. Tian, "A survey on load balancing algorithms for virtual machines placement in cloud computing," no. February, pp. 1–16, 2017.

[6] Y. W. Qiu and J. I. G. Hwang, "A Two-Level Load Balancing Method with Dynamic Strategy for Cloud Computing," *Proc. - 2016 IEEE 14th Int. Conf. Dependable, Auton. Secur. Comput. DASC 2016, 2016 IEEE 14th Int. Conf. Pervasive Intell. Comput. PICom 2016, 2016 IEEE 2nd Int. Conf. Big Data Intell. Comput. DataCom 2016 2016 IEEE Cyber Sci. Technol. Congr. CyberSciTech 2016, DASC-PICom-DataCom-CyberSciTech 2016*, pp. 565–571, 2016.

[7] N. K. Chien, N. H. Son, and H. Dac Loc, "Load balancing algorithm based on estimating finish time of services in cloud computing," *2016 18th Int. Conf. Adv. Commun. Technol.*, pp. 1–1, 2016.

[8] U. Thakkar, "A Novel Approach for Dynamic Selection of Load Balancing Algorithms in Cloud Computing," pp. 1–4, 2016.

[9] Y. Gao, "Energy-aware Load Balancing in Heterogeneous Cloud Data Centers."

[10] S. G. Damanal and G. R. M. Reddy, "Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines," 2014.

[11] S. G. Domanal and G. R. M. Reddy, "Load Balancing in Cloud Computing Using Modified Throttled Algorithm."

[12] D. Malhotra, Bhagyalakshmi, "A Review : Different Improvised Throttled Load Balancing Algorithms in Cloud Computing Environment," *Int. J. Eng. Technol. Manag. Appl. Sci.*, vol. 5, no. 7, pp. 409–416, 2017.

[13] R. Panwar and B. Mallick, "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm," *Ieee*, pp. 773–778, 2015.

[14] S. Elmougy, S. Sarhan, and M. Joundy, "A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique," *J. Cloud Comput. Springer*, 2017.

Authors Profile

Ms. Bhagyalakshmi pursed Bachelor of Computer Science from University of Jammu, India in 2010 and Master of Computer Science and Engineering from Shri Mata Vaishno Devi University, India in year 2013. She is currently pursuing Ph.D. and from the Department of Computer Science & Information Technology, School of Applied Sciences since 2016. Her main research work focuses on routing in omputer networks and cloud computing.

Dr Deepti Malhotra pursed Bachelor of Computer Science from University of Jammu, India in 2005, M.E. Computer Science from Thappar University, India in year 2007 and completed her Ph.D. from University of Jammu in 2013. She is currently working as Assistant Professor in Department of CS&IT in Central University of Jammu, J&K, India. She has published more than 15 research papers in reputed international journals and conferences. Her main research work focuses on Grid Computing, Cloud Computing, Big Data Analytics and Data Mining.

