American Journal of Engineering Research (AJER) ***2019***
American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-8, Issue-4, pp-99-105
www.ajer.org
Research Paper Open Access

Load Balancing Approach and Algorithm in Cloud Computing Environment

Marzana Ifat Moly, Md. Alomgir Hossain, Senior Lecturer, Ovijit Roy,
Department of Computer Science and Engineering
IUBAT – International University of Business Agriculture and Technology
Embankment Drive Road, Sector-10, Uttara Model Town, Dhaka-1230
Corresponding Author: Marzana Ifat Moly

ABSTRACT: *Cloud computing is a new technology that brings new challenges to all organizations around the world. Improving response time for user requests on cloud computing is a critical issue to combat bottlenecks. It has a number of types and hybrid cloud is one of them. Delivering services in a hybrid cloud is an uphill task. One of the challenges associated with this paradigm is the even distribution among the resources of a hybrid cloud, often refereed as load balancing. In cloud computing load balancing is a key issue. It would consume a lot of cost to maintain load information. Good load balancing makes cloud computing more efficient and improves user satisfaction. Many algorithms were suggested to provide efficient mechanisms and algorithms for assigning the client's requests to available Cloud nodes. We use the Round Robin algorithm in this paper and try to find the load balancing in the cloud computing environment.*
KEYWORDS: *Load Balancing, Cloud Computing, Round Robin Algorithm, Scheduling.*

Date of Submission:28-03-2019 Date of acceptance: 08-04-2019

I. INTRODUCTION

Cloud computing is a recent technology that concern with online distribution of computing resources and services. In cloud computing, end-user knowledge about the configuration of service delivering system may not be required because client just use services on pay per model where all system configuration and resource management is taken care by cloud system automatically [1]. One important issue associated with this field is dynamic load balancing or task scheduling. Load balancing algorithms were investigated heavily in various environments. Cloud Computing the main concerns involve efficiently assigning tasks to the Cloud nodes such that the effort and request processing is done as efficiently as possible, while being able to tolerate the various affecting constraints such as heterogeneity and high communication delays [2].

II. LITERATURE REVIEW

In a cloud environment, each host as a computational node performs a task or a subtask. The Opportunistic Load Balancing algorithm (OLB) intends to keep each node busy regardless of the current workload of each node [3]. OLB assigns tasks to available nodes in random order. The Minimum Completion Time algorithm (MCT) assigns a task to the node that has the expected minimum completion time of this task over other nodes [4]. Cloud computing has emerged as a buzzword in the commercial and academic world, for its great potential to fulfill the envisioned blueprint that customers can enjoy computing infrastructure and services in a pay-as-you-go manner[5]. Generally clouds give customers three levels of access: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructureas-a-Service (IaaS)[6]. Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers [7]. Cloud computing is an emerging computing model based on the development of distributed computing, parallel processing and grid computing [8].

III. CLOUD COMPUTING

Cloud computing is a utility to deliver services and resources to the users through high speed internet. It has gained immense popularity in recent years. These cloud computing services can be used at individual or

corporate level. Cloud computing can be summarized as a model that gives access to a pool of recourses with minimal management effort [9].

Types of clouds: Clouds can be classified as private, public and hybrid on the basis of their architecture. It provides three types of services 1. Infrastructure as a Service (IAAS), that provides the infrastructure a user demands like routers. 2. Software as a Service (SAAS), delivers software services like Google Apps. 3. Platform as a Service, PAAS, as the name suggests provides platforms for program development for example Google's App Engine. **Private cloud:** A cloud used only within an enterprise is referred as a private cloud. It can also be addressed as internal cloud. They are managed by the organization itself.

Public cloud: A cloud that is made available to the users around the globe through an Internet access is called a public cloud. Organizations providing such cloud services include Google Docs, Microsoft's Windows Azure Platform, Amazon's Elastic Compute Cloud and Simple Storage Services, IBM's Smart Business Services.

Hybrid cloud: A union of private and public clouds forms another type of cloud referred as hybrid cloud. As one part of it is private, it is considered to be more secure but designing a hybrid cloud is a challenging job because of the complexities involved in the design phase. The major issues linked with them are that of interoperability and standardization. They are costly as compared to the aforementioned types but it has their best features combined [10].

IV. LOAD BALANCING

Load balancing is also one of the main challenges faced in hybrid cloud computing, as there is a need for an even and dynamic distribution of load between the nodes in private and public clouds. In distributed systems load balancing is defined as the process of distributing load among various nodes to improve the overall resource utilization and job response time. While doing so, it is made sure that nodes are not loaded heavily, left idle or assigned tasks lesser then its capacity. It is ensured that all the nodes should be assigned almost the same amount of load. If resources would be utilized optimally, performance of the system will automatically increase. Not only this, the energy consumption and carbon emission will also reduce tremendously. It also reduces the possibility of bottleneck which occurs due to the load imbalance. Furthermore, it facilitates efficient and fair distribution of resources and helps in the greening of these environments. Load balancing algorithms are classified into categories for the ease of understanding. That helps in identifying a suitable algorithm in the time of need. A detailed view of classification is presented below [11].

V. LOAD BALANCING STRATEGIES FOR CLOUDS

Load balancing algorithms can be broadly categorized into static and dynamic load balancing algorithms.
Static load balancing algorithms: Gulati et al. [12] claimed that in cloud environment a lot of work is done on load balancing in homogeneous resources. Research on load balancing in heterogeneous environment is given also under spot light. They studied the effect of round robin technique with dynamic approach by varying host bandwidth, cloudlet long length, VM image size and VM bandwidth. Load is optimized by varying these parameters. CloudSim is used for this implementation.

Dynamic load balancing algorithms: A hybrid load balancing policy was presented by Shu-Ching et al. [13]. This policy comprises of two stages 1) Static load balancing stage 2) Dynamic load balancing stage. It selects suitable node set in the static load balancing stage and keeps a balance of tasks and resources in dynamic load balancing stage. When a request arrives a dispatcher sends out an agent that gathers nodes information like remaining CPU capacity and memory. Hence the duty of the dispatcher is not only to monitor and select effective nodes but also to assign tasks to the nodes accordingly. Their results showed that this policy can provide better results in comparison with min-min and minimum completion time (MCT), in terms of overall performance. Another algorithm for load balancing in cloud environment is ant colony optimization (ACO). This work basically proposed a modified version of ACO. Ants move in forward and backward directions in order to keep track of overloaded and under loaded nodes. While doing so ants update the pheromone, which keeps the nodes' resource information. The two types of pheromone updates are 1) Foraging pheromone, which is looked up when an under loaded node is encountered in order to look for the path to an over loaded node. 2) Trailing pheromone is used to find path towards an under loaded node when an over loaded node is encountered. In the previous algorithm ants maintained their own result sets and were combined at a later stage but in this version these result sets are continuously updated. This modification helps this algorithm perform better.

VI. PROPOSED SYSTEM

Round Robin Algorithm

The round-robin (RR) planning calculation is composed particularly for time-sharing systems. It is like FCFS planning, yet pre-emption is further to switch between procedures. A little unit of time, called a time quantum or time cut, is distinctive. The prepared line is considered as a roundabout line. To perform RR booking, we keep the prepared line as a FIFO line of procedures. New procedures are added beside the prepared line. The CPU scheduler picks the primary procedure from the prepared line, plans a time quantum to hinder after some time, and dispatches the procedure. The procedure may have a CPU burst not exactly the time quantum. For this situation, the procedure itself will release the CPU willfully. The scheduler will then continue to the following procedure in the prepared line. Something else, if the CPU burst of the at present running procedure is longer than the given time quantum, the time will go off and will bring about a hinder to the OS. A context switch will be executed, and the procedure will be put at the tail of the prepared line. The CPU scheduler will then choose the following procedure in the readied line.

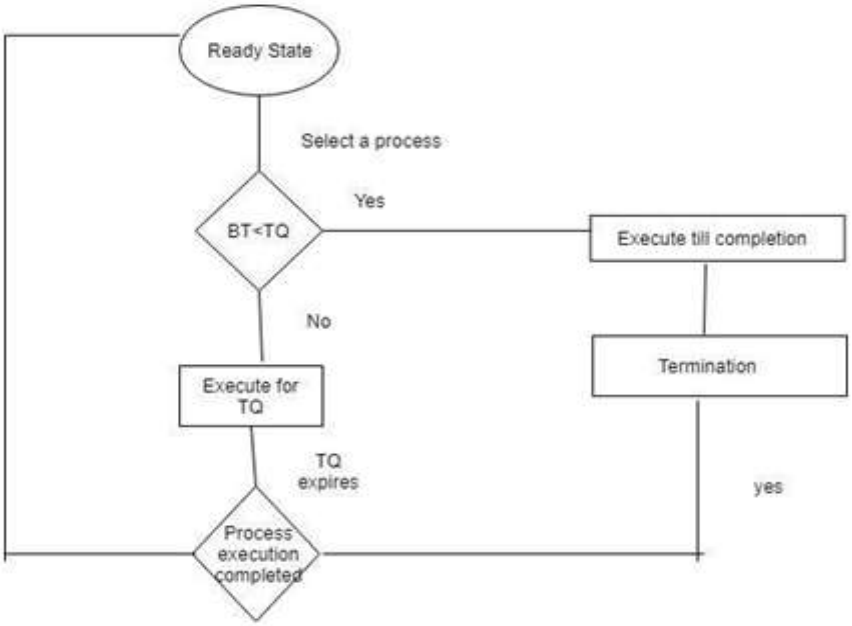

Fig 1: Flow Chart

We assume five processes arriving at time=0, with increasing burst time (P1=4, P2=5, P3=2, P4=1, P5=6, P6=3)

Process	Arrival Time	Burst Time
P1	0	4
P2	1	5
P3	2	2
P4	3	1
P5	4	6
P6	6	3

Fig 2: Example of Round Robin Algorithm

The Gantt chart is shown below:

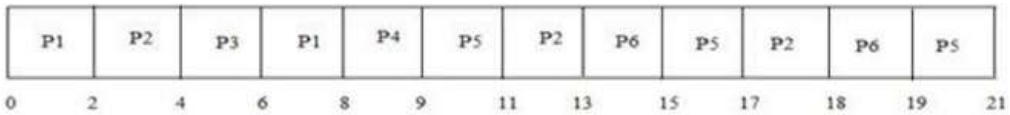

Fig 3: Gantt Chart

Average Waiting Time= 8.04
Average Turn Around Time= 11.16
Context Switching (if Time Quantum decreases CS increases and vice versa) = 7

VII. METHOD

The progress of Round Robin is presented is as following:
Step 1: It is to calculate the average completion time of each task for all nodes, respectively.
Step 2: It is to find the task that has the maximum average completion time.
Step 3: It is to find the unassigned node that has the minimum completion time less than the maximum average completion time for the task selected in Step 2. Then, this task is dispatched to the selected node for computation. **Step 4**: If there is no unassigned node can be selected in Step 2, all nodes including unassigned and assigned nodes should be reevaluated. The minimum completion time of an assigned node is the sum of minimum completion time of assigned task on this node and the minimum completion time of the current task. The minimum completion time of an unassigned node is the current minimum completion time for the task. It is to find the unassigned node or assigned node that has the minimum completion time less than the maximum average completion time for the task selected in Step 2. Then, this task is dispatched to the selected node for computation.
Step 5: Repeat Step 2 to Step 4, until all tasks have been completed totally. In the following section, an example to be executed by using the proposed algorithm is given.

VIII. PROPOSED ALGORITHM

In our proposed formula, Modified spherical Robin formula. The number of processes is residing within the prepared queue, we assume their arrival time is assigned to some processes and burst times are allotted to the computer hardware. The burst time and the number of processes are thought-about as input. Now initial of all we tend to organize all processes in increasing order according to their given burst time and opt for changed time slice are going to be depends on the quantity of processes burst time. If number of processes are varied then time slice can vary. In this algorithm some variety of processes and their burst time are given. And we have to be compelled to determine their context switch time their turnaround and their waiting time. In modified spherical robin formula shortest job initial and spherical robin algorithms are mixed up. So that this formula will take the benefits of each formula. And execute more expeditiously.

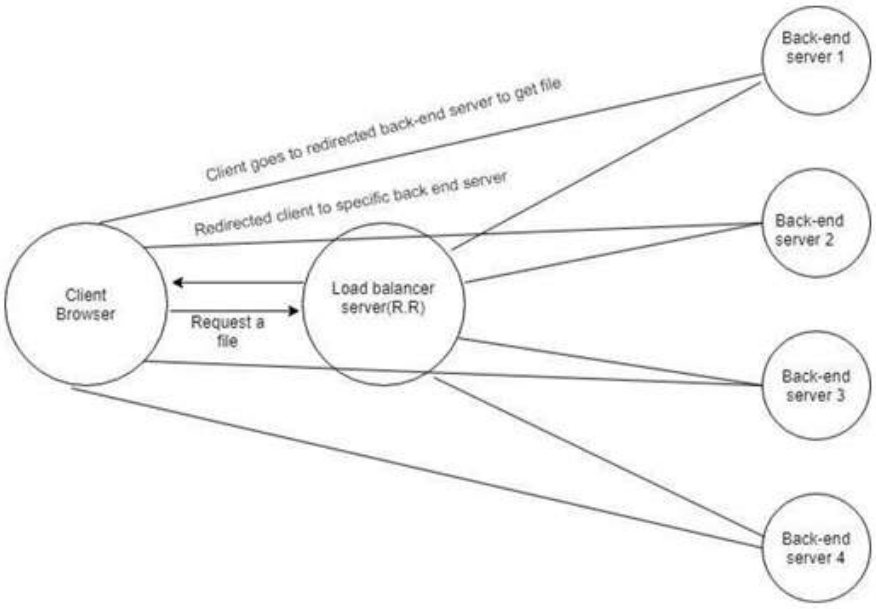

Fig 4: Load Balancer Server (R.R)

Steps for Round Robin Formula
Step 1: first check the standing of prepared queue.
Step 2: change the standing of all the processes to prepared state.
Step 3: arrange all the processes in increasing order or type processes according to their burst time.
Step 4: check whether prepared queue is null or not.
Step 5: if not then calculate modified time quantum Time Quantum= No of Burst times/ No of processes. Step 6: assign modified time quantum to each method.
Step 7: if burst time of method is smaller and equal to time quantum then process complete its execution. Otherwise repeat cycle and give the time quantum to every method.
Step 8: repeat step 6 till all processes will not complete their execution.

Process	Arrival Time	Burst Time
P1	0	4
P2	1	5
P3	2	2
P4	3	1
P5	4	6
P6	6	3

Fig 5: Round Robin Algorithm

Quantum= AVG. of Burst time/No of processes
Time Quantum= 4

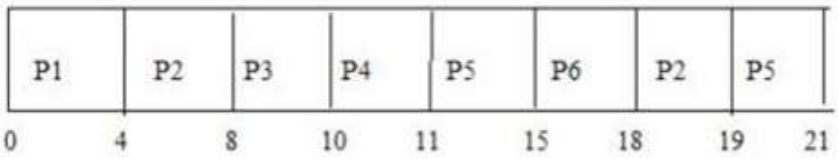

Fig 6: Gantt Chart

Average Waiting Time= 7.56
Average Turnaround Time= 10.833
Context Switching= 6

IX. COMPARISON

A comparative study of different load balancing algorithms is presented in. Load balancing is not only required for meeting users' satisfaction but it also helps in proper utilization of the resources available. The metrics that are used for evaluating different load balancing technologies are: throughput, overhead associated, fault tolerance, migration time, response time, resource utilization, scalability, and performance. According to this study, in honeybee foraging algorithm, throughput does not increase with the increase in system size. Biased random sampling and active clustering do not work well as the system diversity increases. OLB + LBMM shows better results than the algorithms listed so far, in terms of efficient resource utilization. The algorithm Join-Idle-Queue can show optimal performance when hosted for web services but there are some scalability and reliability issues that make its use difficult in today's dynamic-content web services. They further added that min min algorithm can lead to starvation. They concluded that one can pick any algorithm according to ones needs. There is still room for improvement in all of these algorithms to make them work more efficiently in heterogeneous environments while keeping the cost to a minimum. A somewhat similar analysis of load balancing algorithms is presented by Daryapurkar et al. And Rajguru and Apte as well. Different scheduling algorithms for the hybrid clouds compared by Bittencourt et al. [14], highlights that the maxspan of these algorithms widely depend on the bandwidth provided between the private and public clouds. The channels are usually part of the internet backbone and their bandwidth fluctuates immensely. This makes the designing of the communication aware algorithms quite challenging.

Round Robin Vs Weighted Round Robin

Round Robin is undoubtedly the most widely used algorithm. It's easy to implement and easy to understand. Here's how it works. Let's say you have 2 servers waiting for requests behind your load balancer. Once the first request arrives, the load balancer will forward that request to the 1st server. When the 2nd request arrives (presumably from a different client), that request will then be forwarded to the 2nd server.
Because the 2nd server is the last in this cluster, the next request (i.e., the 3rd) will be forwarded back to the 1st server, the 4th request back to the 2nd server, and so on, in a cyclical fashion.

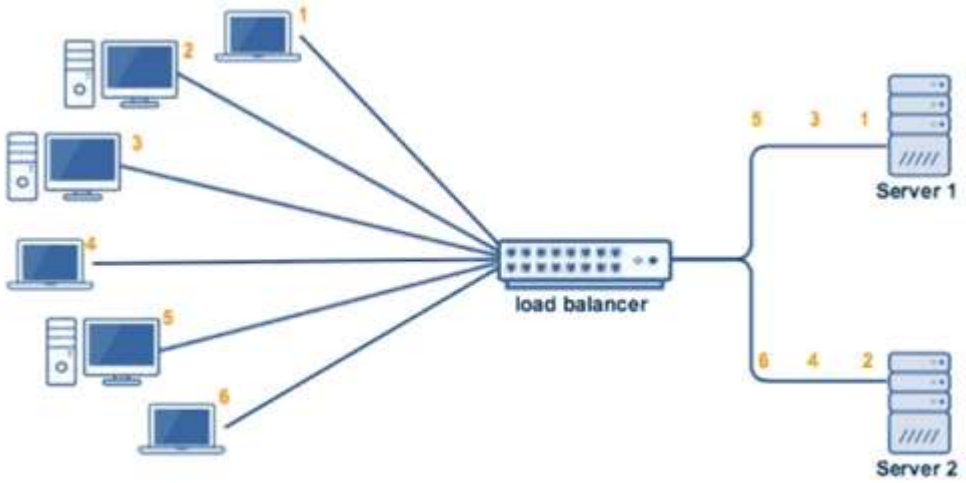

Fig 7: Round Robin Load Balancer

The Weighted Round Robin is similar to the Round Robin in a sense that the manner by which requests are assigned to the nodes is still cyclical, albeit with a twist. The node with the higher specs will be apportioned a greater number of requests.

But how would the load balancer know which node has a higher capacity? Simple. You tell it beforehand. Basically, when you set up the load balancer, you assign "weights" to each node. The node with the higher specs should of course be given the higher weight.

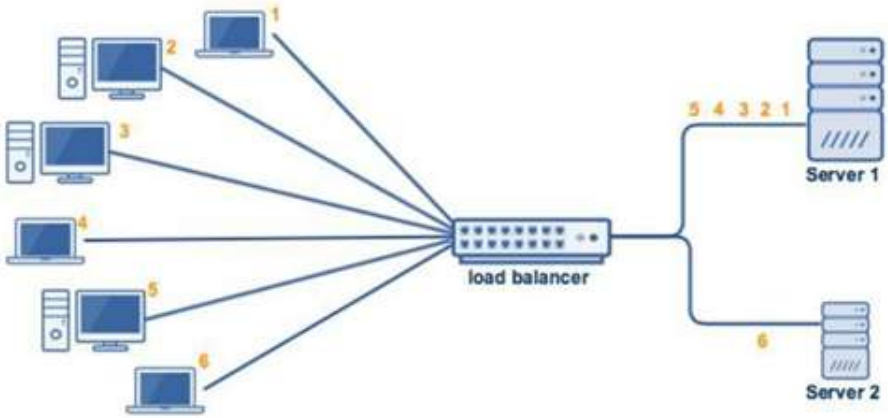

Fig 8: Weighted Round Robin Load Balancer

X. CONCLUSION

In this paper, we proposed an efficient Round Robin algorithm, for the cloud computing network to assign tasks to Load Balancing according to their resource capability. Similarly, Round Robin Algorithm can achieve better load balancing and performance than other algorithms, such as Weighted Round Robin and Shortest Job First from the study. One of the major issues of cloud computing is load balancing because overloading of a system may lead to poor performance which can make the technology unsuccessful. So there is always a requirement of efficient load balancing algorithm for efficient utilization of resources.

REFERENCE

[1]. Yatendra Sahu," Cloud Server Optimization with Load Balancing and Green Computing Techniques Using Dynamic Compare and Balance Algorithm", 2013 5th International Conference on Computational Intelligence and Communication Networks 978 -0-7695-5069-5/13 © 2013

[2]. IEEE

[3]. Klaithem Al Nuaimi," A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", 2012 IEEE Second Symposium on Network Cloud Computing and Applications 978-0-7695-4943-9/12 © 2012 IEEE

[4]. Freund, R. F., Siegel, H. J. : Heterogeneous processing. IEEE Computer, vol. 26, pp.13— 17, (1993)

[5]. Ritchie, G., Levine, J.: A Fast, Effective Local Search for Scheduling Independent Jobs in Heterogeneous Computing Environment s. Journal of Computer Applications, vol. 25, pp. 1190—1192, (2005)

[6]. Zheng Hu,"An Utility- Based Job Scheduling Algorithm For Current Computing Cloud Considering Reliability Factor".

[7]. Jeffrey Galloway," An Empirical Study of Power Aware Load Balancing in Local Cloud Architectures", 2012 Ninth International Conference on Information Technology- New Generations 978-0-7695-4654-4/12 © 2012 IEEE

[8]. Gaochao Xu," A Load Balancing Model Based on Cloud Partitioning for the Public Cloud", TSINGHUA SCIENCE AND TECHNOLOGY ISSNl l1007-0214, 2013

[9]. Haozheng Ren," The Load Balancing Algorithm in Cloud Computing Environment",2012 2nd International Conference on Computer Science and Network Technology 978-1-4673-2964-4/12 ©2012 IEEE

[10]. https://www.google.co.in/docs/

[11]. Adler B (2012) Designing Private and Hybrid Clouds. Architectural Best Practices.

[12]. Gulati A, Chopra RK (2013) Dynamic Round Robin for Load Balancing in a Cloud Computing, International Journal of Computer Science and Mobile Computing 2: 274-278.

[13]. Wang SC, Chen CW, Yan KQ, Wang SS (2013) The Anatomy Study of Load Balancing in Cloud Computing Environment. The Eighth International Conference on Internet and Web Applications and Services 230-235.

[14]. Bittencourt LF, Madeira ERM, Fonseca N (2012) Scheduling in hybrid clouds. Communications Magazine, IEEE.

[15]. H. S. Behera, R. Mohanty, and D. Nayak, A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis, (2010), Vol.5, No.5, pp.10-15.

[16]. S. M. Mostafa, S. Z. Rida, and S. H. Hamad, Finding Time Quantum of Round Robin CPU Scheduling Algorithm in General Computing Systems using Integer Programming, IJRRAS, (2010), Vol.5, No.1, pp.64-71.

Marzana Ifat Moly" Load Balancing Approach and Algorithm in Cloud Computing Environment" American Journal of Engineering Research (AJER), vol.8, no.04, 2019, pp.99-105