International Journal of Communication and Networking System
Volume: 06 Issue: 01 June 2017 Page No.1- 4
ISSN: 2278-2427

Efficient Static Load Balancing Algorithm in Cloud Computing

Senorita Sundarrajan, Sherina Sundarrajan, Varshini. M
CSE, St.Joseph's College of Engineering, Chennai, India
Email: senoritasundarrajan@gmail.com, sherinasundarrajan@gmail.com, varshini.m.13@gmail.com

Abstract — Cloud Computing is one of the emerging necessities in the IT world and has shown immense growth over the years. The applications can be run remotely using other people's servers. This is done with a simple user interface or an application format. In general, Cloud is simply the Internet, which in turn is a network of remote servers. "Payment only for what you are using" is the bottom line for computing in cloud environment. The Cloud is a pool of heterogeneous resources. The storage dealt with cloud computing has opened the gates to Load Balancing. The prime motive of this paper is to make efficient use of the existing static load balancing algorithms. This is implemented by the addition of Max-Min Algorithm and Weighted Round Robin Algorithm, the efficient and the successful division of the workload, thereby multiplying the scalability and the performance of computing, subtracting the response time.

Keywords— Cloud, Load Balancing, Max-Min, Weighted Round Robin.

I. INTRODUCTION

A cloud computing system has data centers, clients and distributed servers as its main ingredients. Instead of managing one's own data, one's own software and one's own server on premise, which simply means managing an entire stack, the cloud allows one to outsource some of those.

II. STATEMENT OF THE PROBLEM

Crashing of servers is one amongst the serious problems faced in online websites today. Although several alternatives has been proposed to reduce this problem of overloading servers, an efficient method has not yet been proposed in order to overcome the problem completely. The objectives involved include balancing the load efficiently. The prime task is to check for the services and to increase its availability. The Cloud offers innumerable services and resources to the users. The virtual and the cloud computing environments help the organizations to enjoy various benefits where flexibility, scalability and accessibility top the list. The paper is concerned in maximizing the utilization of resources, thereby increasing the satisfaction of the users. The time taken to wait and the time involved in executing a particular task must be reduced. The complex cloud software stacks and the various technologies involved in virtualization, may lead to countless bugs and inefficiencies. The reliability of the systems and the servers must be ensured to bring about efficiency. The essence of the overall aim and ambition of the paper is to build a fault tolerant system. The failure of one particular node must not result in disrupting the entire system. The provisioning of services should not be affected due to a single point of failure that is said have occurred. The challenges involved in load balancing incorporate scalability where the quality of the service rendered should remain the same even after the addition of huge number of users. The task, once submitted must yield in quick results thus targeting faster response time. Reacting to a particular request quickly and effectively is the supreme goal of balancing load in cloud computing. The purpose of using a cloud environment is to reduce the cost involved in buying new servers for every single machine. Therefore, the overhead associated such as the indirect costs or burdens must be diminished greatly. Although the services provided, prove to be of immense nourishment to the job, the servers' performance seems to be uncertain. The ultimate target is to bring the performance of the system to the peak. The idea of the paper skates to accommodate future modifications, if any.

III. OBJECTIVE OF THE STUDY

1. To analyse the blue print of cloud computing.
2. To probe into an assortment of algorithms that have been proposed already.
3. To furnish an optimal solution based on the assortment of algorithms.

IV. CLOUD COMPUTING – AN OVERVIEW

A cloud computing system has data centers, clients and distributed servers as its main ingredients. Instead of managing one's own data, one's own software and one's own server on premise, which simply means managing an entire stack, the cloud allows one to outsource some of those. All the storage here has considered self-service and the user is responsible for updating, downloading and managing their storage by themselves. Cloud is classified based on its Capability and its Accessibility.[1]

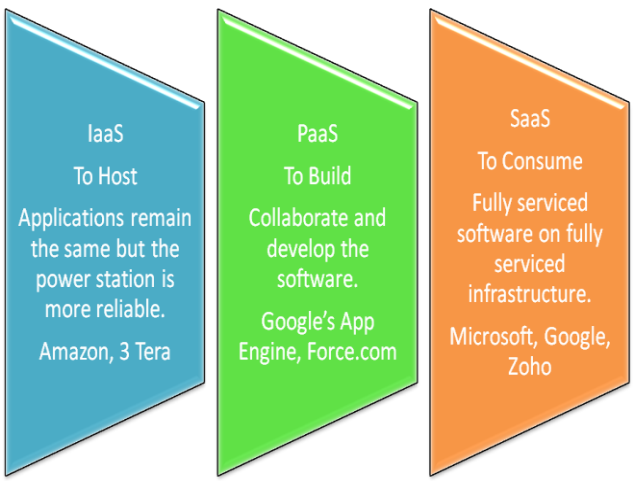

Fig. 1. CAPABILITY- SERVICE MODELS

International Journal of Communication and Networking System
Volume: 06 Issue: 01 June 2017 Page No.1- 4
ISSN: 2278-2427

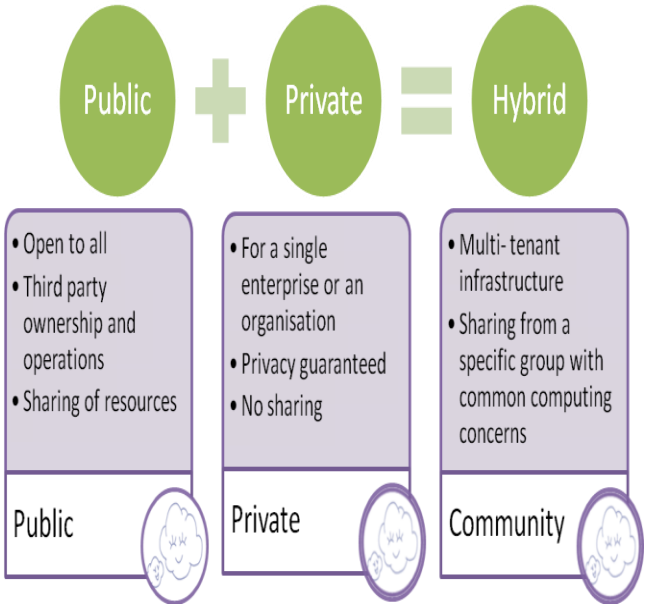

Fig. 2. THE ACCESSIBILITY - DEPLOYMENT MODELS

V. LOAD BALANCING

Load Balancing refers to the uniform distribution of excess workload among all the nodes in the network. The objectives and goals include maximizing utilization of resources, considering satisfaction of users, maintaining system stability by building fault tolerant systems.[2]

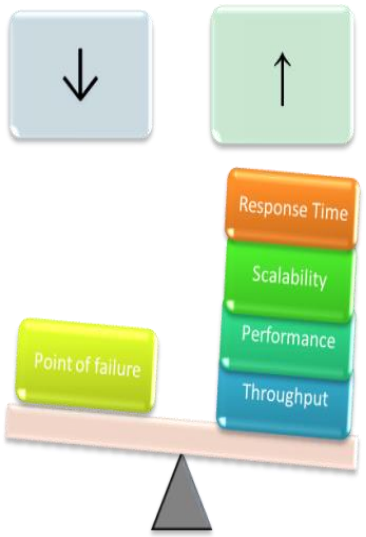

Fig. 3. LOAD BALANCING

VI. STATIC LOAD BALANCING

It depends on the prior knowledge and does not need to know the current status of the node.[3] The node's performance is determined when the execution commences. The allotted work is calculated by the nodes and they finally submit their result to the remote node. Without the current load being considered the workload is evenly distributed depending on the performance. Non - preemptive task execution is followed in the static load balancing methods where the load once allocated to one particular node cannot be transferred to another node, thereby requiring less communication reducing the execution time.

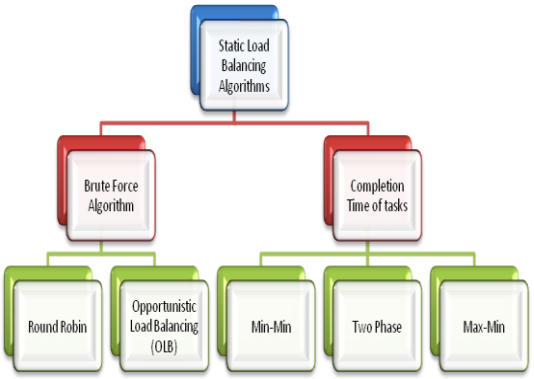

FIG. 4. CATEGORIES OF STATIC LOAD BALANCING ALGORITHMS

VII. A PROBE INTO VARIOUS STATIC LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING

A. Round Robin Algorithm

Round Robin algorithm works best for clusters consisting of servers with identical specifications. It chooses a node randomly and allocates the task in a circular fashion.[4] Even though it is the simplest method for distributing the request sent by the clients across a group of servers, round robin algorithm suffers non- uniformity in distributing the workload due to the specifications of the servers being identical.

B. Weighted Round Robin

To overcome the obstacles that hindered the growth of Round Robin algorithm, Weighted Round Robin Algorithm was proposed in which the servers with higher specifications and capabilities are given more weightage than the others. Each node is assigned with weights based on the specifications of the servers according to which the client requests are served.[5]The server receives larger proportion of client requests if they weigh more. For instance, if server A is assigned weight 5 and B is assigned weight 1, load balancer forwards 5 requests to A for each 1 it sends to B. It plays a vital role in running business critical applications where the server gets substantially lower number of connections than an equally capable server.

C. Opportunistic Load Balancing Algorithm

The main aim of this algorithm is to make sure that every node is busy and does not remain idle at any point of time. The tasks that need to be executed are assigned in a random order to the node that seems to be currently available.[6] Since the current execution time of the nodes are unknown, the processing speed is greatly reduced generating poor make span.

D. Min-Min Load Balancing Algorithm

The parameters that are required to be known in advance for the tasks, which are waiting in a queue include execution time and completion time. The unexecuted tasks with the minimum execution time are first assigned to the processors and the rest are handled afterward. This is done in accordance with the capability of job completion in a specified completion time.[7] The unexecuted tasks might experience starvation while they are already waiting in the queue for a long time.

E. Two Phase Load Balancing Algorithm

The Two Phase Load Balancing Algorithm is a combination of Opportunistic Load Balancing (OLB) and Load Balancing

International Journal of Communication and Networking System
Volume: 06 Issue: 01 June 2017 Page No.1- 4
ISSN: 2278-2427

Min-Min (LBMM). The primary goal is to achieve resource utilization and enhance work efficiency for which this algorithm works fine. As explained earlier, OLB ensures that every node is kept busy and LBMM takes the burden of minimizing the execution time, reducing the overall completion time as a result.[8]

F. Max-Min Load Balancing Algorithm

This algorithm deals with assigning tasks with longer execution time to the best servers first, along with which the rest of the smaller tasks to be executed are assigned to all other servers. This ensures concurrent execution of tasks and does not suffer starvation. Thus, it serves with minimum completion time improving make span.[9] For example, task A from the meta task is assigned to the best server as it takes longer execution time than the others whilst task B and C comparatively having lower execution time than task A and is given to the other servers. Thus, the smaller tasks B and C are executed concurrently along with task A.

TABLE 1: COMPARATIVE STUDY OF EXISTING ALGORITHMS

	ROUND ROBIN	OLB	MIN-MIN	TWO-PHASE	MAX-MIN
THROUGHPUT	YES	NO	YES	YES	YES
OVERHEAD	YES	NO	YES	YES	YES
FAULT TOLERANCE	NO	NO	NO	NO	NO
RESPONSE TIME	YES	NO	YES	YES	YES
RESOURCE UTILIZATION	YES	YES	YES	YES	YES
SCALABILITY	NO	NO	NO	NO	NO
PERFORMANCE	YES	YES	YES	YES	YES

SOURCE: Abhijit Aditya, Uddalak Chatterjee and Snehasis Gupta, A Comparative Study of Different Static and Dynamic Load Balancing Algorithm in Cloud Computing with Special Emphasis on Time Factor [10]

VIII. PROPOSED SYSTEM

The paper centers two existing algorithms serving their purpose more efficiently in combinations. The algorithms considered are the Max- Min algorithm and the Weighted Round Robin algorithm. In the Max-Min Algorithm, the task with the maximum execution time is first assigned to the processors with maximum capacity and the tasks with minimum execution time are processed concurrently by other processors. The capacity of the servers can be determined using weighted round robin. This combination of these algorithms helps reduce the response time and increase the throughput thereby increasing the overall performance of load distribution.

A.Weighted Round-Robin Algorithm

Let $S = \{S_1, S_2, \ldots\ldots\ldots S_n\}$ be the servers.

Let $w = \{w_1, w_2, \ldots\ldots\ldots w_n\}$ be the weights assigned to the servers.

Let C be the number of tasks assigned to the processors.

{

If w_1 = Maximum weight

{

Assign task T_1 to server with weight with w_1

C++;

If (C > weight w_1)

{

Assign task to the next server with maximum weight

}

If (S==NULL)

{

Determine Completion time

}

}

B.Max-Min Algorithm

For all the submitted tasks T_i

{

Resources assigned is R_j

Processor P_1 = maximum capacity

If Task T_i = maximum execution time

{

Assign Processor P_1 to T_i

Remove task T_i from the set

Update the resource set R_j

}

If (Tasks (T_1 , T_2 ,.....T_k) = T_k AND T_k = minimum execution time)

{

Assign T_k simultaneously to other processors

}

Update the resource set

Determine the Completion Time CT_k

}

IX. CONCLUSION

The load balancer serves like a traffic cop glancing over all the challenges faced. It routes client requests to servers in such a way that neither the servers are overloaded, nor the servers remain idle. The principle aim of this paper is to take into consideration suitable parameters and suggest an appropriate method in order to overcome the problems faced while balancing the load. This combination of the Weighted Round Robin and the Max-Min algorithms helps reduce the response time and increase the throughput thereby increasing the overall performance of load distribution.

References

[1] Christof Weinhardt, Arun Anandasivam, Benjamin Blau, Nikolay Borissov, Thomas Meinl, Wibke Michalk and Jochen Stober, Cloud Computing – A Classification, Business Models, and Research Directions, Business & Information Systems Engineering, 2009.

[2] Neeraj Rathore and Inderveer Chana, Load Balancing and Job Migration Techniques in Grid: A Survey of Recent Trends, Wireless Personal Communications, 2014.

[3] Kwang Mong Sim and Weng Hong Sun, Ant colony optimization for routing and load-balancing: survey and new directions, IEEE, 2003.

[4] Ankush P. Deshmukh and Kumaraswamy Pamu, Applying Load Balancing: A Dynamic Approach, International Journal of Advanced Research in Computer Science and Software Engineering, 2012.

[5] James, Jasmin and Verma, Bhupendra, Efficient VM load balancing algorithm for a cloud computing environment,

International Journal of Communication and Networking System
Volume: 06 Issue: 01 June 2017 Page No.1- 4
ISSN: 2278-2427

International Journal on Computer Science and Engineering, 2012.

[6] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao and Shun-Sheng Wang, Towards a Load Balancing in a three-level cloud computing network, 3rd International Conference on Computer Science and Information Technology, IEEE, 2010.

[7] T. Kokilavani and D.I. George Amalarethinam, Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing, International Journal of Computer Applications, 2011.

[8] G. Attiya and Y. Hamam, Two phase algorithm for load balancing in heterogeneous distributed systems, 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, IEEE, 2004.

[9] TD Braun, HJ Siegel, N Beck and LL Boloni, A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems, Journal of Parallel and Distributed Computing, 2001.

[10] Abhijit Aditya, Uddalak Chatterjee and Snehasis Gupta, A Comparative Study of Different Static and Dynamic Load Balancing Algorithm in Cloud Computing with Special Emphasis on Time Factor, International Journal of Current Engineering and Technology, 2015.