Int. J. Advanced Networking and Applications
Volume: 11 Issue: 01 Pages: 4177-4183(2019) ISSN: 0975-0290

Features Exploration of Distinct Load Balancing Algorithms in Cloud Computing Environment

Saurabh Gupta
Research Scholor,UTU, Dehradoon
Email: saurabh256837@gmail.com
Sangeeta Rani
Assistant Professor , SGT University, Gurugram
Email: sangeeta.sept@gmail.com
Dr. Amit Dixit
Dean and Professor, Quantum University, Roorkee
Email: dixitamit777@gmail.com
Dr. Harsh Dev
Dean and Professor, PSIT, Kanpur
Email: drharshdev@gmail.com

ABSTRACT

The delivery of Cloud computing is a method for delivering information /services in which resources are retrieved from the Internet through web-based tools and applications, as opposed to a direct connection to a server rather than keeping files on a proprietary hard drive or local storage device. We have studied a lot of algorithm for reducing the response time of load balancing algorithm in cloud computing environment. We also measured the execution time of different algorithms. In this paper we will compare execution time, processing time of data center, response time of various algorithms.

Date of Submission: May 16, 2019 Date of Acceptance: June 17, 2019

Introduction

In the computer system, a cloud is a storage database where there are lots of servers. Hence, the users can save their data securely. Load balancing is needed to balance the load between all the servers. Load balancing achieves the balancing of the servers by moving the processes or tasks to lightly loaded servers from overloaded servers so that it increases the performance of the system.[3] Cloud computing has made drastic change in the reduction of cost of hardware and software and other server resources as well. With the help of cloud we can access any data, applications whenever and wherever we want to, over the internet. There are six major reasons why we use cloud services.

- Maintaining focus on the Business
- Business Agility
- Reduced Capital Expenditures
- Scale
- Anytime access from anywhere
- Staffing Efficiency

Load balancing

Load balancing is the mechanism to balance the load to the Cloud nodes in a manner that
Each & every node can utilize the resources and minimize response time as well. It also gives solution to the problem in which some cloud providers are over loaded or some cloud providers are under loaded or remain idle.
There are number of techniques and algorithms that can be used to balance load in different nodes. The technique chosen will depend on the type of service or application being served and the status of the network and servers at the time of the request. Load balancing policy can be initiated by the sender or receiver is called sender initiated or server initiated load balancing policy. [4]
If concept of load balancing is predefined and not changing according to situation then it is called static load balancing algorithm. If load balancing approach is changing itself according to situation, then it is called dynamic load balancing approach. [4]

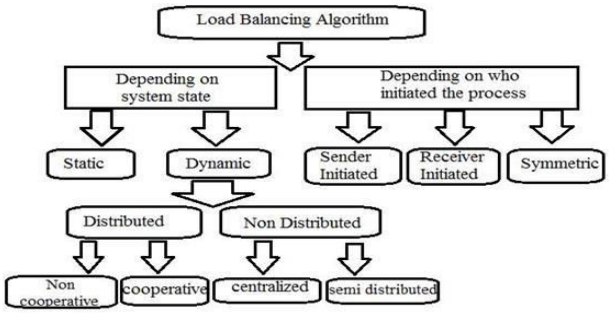

Fig: Types of Load Balancing [4]

1.) Min-Min Load Balancing Algorithm

The cloud manager identifies the execution and completion time of the unassigned tasks waiting in a queue. This is static load balancing algorithm so the parameters related to the job are known in advance. In this type of algorithm the cloud manager first deals with the jobs having minimum execution time by assigning them to the processors according to the capability of complete the job in specified completion time. The jobs having maximum execution time has to wait for the unspecific

Volume: 11 Issue: 01 Pages: 4177-4183(2019) ISSN: 0975-0290

period of time. Until all the tasks are assigned in the processor, the assigned tasks are updated in the processors and the task is removed from the waiting queue. [10]
It begins with a set of all unassigned tasks. First of all, minimum completion time for all tasks is found. Then among these minimum times the minimum value is selected which is the minimum time among all the tasks on any resources. Then according to that minimum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. Then again the same procedure is followed until all the tasks are assigned on the resources. But this approach has a major drawback that it can lead to starvation. [11]

Advantages:

- Easy to implement.
- Dynamic in nature as it considers the VM capacity, task size and present load on the VM.
- This algorithm performs betters when the numbers of jobs having small execution time is more then the jobs having large execution time.

Disadvantages

- The main drawback of the algorithm is that it can lead to starvation.
- Centralized in nature.
- Task buffering results in increase of response time.
- Not ideal when a large number of tasks are expected to arrive.

2.) Max-Min Load Balancing Algorithm

Max-Min is almost same as the min-min algorithm except the following: after finding out minimum execution times, the maximum value is selected which is the maximum time among all the tasks on any resources. Then according to that maximum time, the task is scheduled on the corresponding machine. [10]Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. The assigned task is removed from the list of the tasks that are to be assigned to the processor and the execution time for all other tasks is updated on that processor. Because of its static approach the requirements are known in advance then the algorithm performed well. An enhanced version of max min algorithm was proposed. It is based on the cases, where meta-tasks contain homogeneous tasks of their completion and execution time, improvement in the efficiency of the algorithm is achieved by increasing the opportunity of concurrent execution of tasks on resources. [11]

Advantages:

- Easy to implement.
- Dynamic in nature as it considers the VM capacity, task size and present load on the VM.
- Gives better schedules if few numbers of minimum execution tasks are present.
- Max-min strategy resolves the difficulty of Min-min, by giving Priority to large tasks. The Max-min algorithm selects the task with the Maximum completion time and assigns it to the resource on which achieves minimum execution time. It is clear the Max-min seems better choice whenever the number of small tasks is much more than large ones.

Disadvantage:

One of the features of the Max-min strategy is that chooses large tasks to be executed firstly, which in turn small task delays for long time.

3.) Ant Colony Load Balancing

It is a load balancing mechanism based on ant colony and complex network theory in an open cloud computing federation. Aim of the ant colony optimization to search an optimal path between the source of food and colony of ant on the basis of their behavior. This approach aims efficient distribution of work load among the node. When request is initialized the ant starts movement towards the source of food from the head node. Regional Load Balancing Node (RLBN) is chosen in Cloud Computing Service Provider (CCSP) as a head node. Ants keep records the every node they visits ant record their data for future decision making .Ant deposits the pheromones during their movement for other ants to select next node The intensity of pheromones can vary on the bases of certain factors like distance of food, quality of food etc. When the job gets successful the pheromones is updated. Each ant build their own individual result set and it is later on built into a complete solution. The ant continuously updates a single result set rather than updating their own result set. By the ant pheromones trials, the solution set is continuously updated. [11]

Advantages:

It uses small-world and scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excellent in fault tolerance and has good scalability hence helps in improving the performance of the system.

- Excellent in fault tolerance
- Good scalability

Disadvantage

- Throughput is less.

4.) Honeybee Foraging Algorithm

The main idea behind the Honeybee Foraging algorithm is derived from the behavior of honeybees. There are two kinds of honeybees: finders and reapers. The finder

Volume: 11 Issue: 01 Pages: 4177-4183(2019) ISSN: 0975-0290

honeybees first goes outside of the honey comb and find the honey sources. After finding the source, they return to the honey comb and do a waggle dance indicating the quality and quantity of honey available. Then, reapers go outside and reap the honey from those sources. After collecting, they return to beehive and does a waggle dance. This dance indicates how much food is left. M. Randles proposed a decentralized honeybee based algorithm for self-organization. In this case, the servers are grouped as virtual server and each virtual server have a process queue. Each server, after processing a request from its queue, calculates the profit which is analogous to the quality that the bees show in their waggle dance. If profit is high, the server stays else, it returns to the forage. This algorithm requires that each node to maintain a separate queue. This computation of profit on each node causes additional overhead. It is a nature inspired decentralized load balancing technique which helps to achieve load balancing across heterogeneous virtual machine of cloud computing environment through local server action and maximize the throughput. The current workload of the VM is calculated then it decides the VM states whether it is over loaded ,under loaded or balanced .according to the current load of VM they are grouped. The priority of the task is taken into consideration after removed from the overload VM which are waiting for the VM .Then the task is schedule to the lightly loaded VM. The earlier removed task are helpful for the finding the lightly loaded VM. These tasks are known as scout bee in the next step. Honey Bee Behavior inspired Load Balancing technique reduces the response time of VM and also reduces the waiting time of task. [11]

Advantages

- Achieves global load balancing through local serve actions.
- Performs well as system diversity increases.

Disadvantage

- The disadvantage of this algorithm is that, it does not show any significant improvement in throughput, which is due to the additional queue and the computation overhead.

5.) Biased Random Sampling load balancing Algorithm

Biased Random Sampling Load Balancing Algorithm is dynamic approach, the network is represented in the form of virtual graph. Each server is taken as a vertex of the node and the in degree represents the available free resources the nodes have. On the basis of the in degree the load balancer allocates the job to the node. The nodes have at least one in degree then load balancer allocates the job to that node. When the job is allocates to the node then the in degree is decrement by one, and it's get incremented again when job gets executed. Random sampling technique is used in the addition and deletion of the processes. The processes are centralized by the threshold value, which indicates the maximum traversal from one node to destination node. The length of traversal is known as walk length. The neighbour node of the current node is selected for the traversal. After receiving the request, load balancer selects a node randomly and compares the current walk length with the threshold value. If the current walk length is equal to or greater than the threshold value, the job is executed at that node. Otherwise, the walk length of the job is incremented and another neighbour node is selected randomly. Whenever a client sends a request to the load balancer, the load balancer allocates the job to the node which has at least one in-degree. Once a job is allocated to the node, the in-degree of that node is decremented by one. After the job is completed, the node creates an incoming edge and increments the in-degree by one.

The addition and deletion of processes is done by the process of random sampling. Each process is characterized by a parameter know as threshold value, which indicates the maximum walk length. A walk is defined as the traversal from one node to another until the destination is found. At each step on the walk, the neighbour node of current node is selected as the next node. In this algorithm, upon receiving the request by the load balancer, it would select a node randomly and compares the current walk length with the threshold value. If the current walk length is equal to or greater than the threshold value, the job is executed at that node. Else, the walk length of the job is incremented and another neighbour node is selected randomly.[11]

Advantages

- Achieves load balancing across all system nodes using random sampling of the system domain.
- Performs better with high and similar population of resources.

Disadvantages

- The performance is degraded as the number of servers increase due to additional overhead for computing the walk length.
- Degrades as population diversity increases.

6.) Active Clustering load balancing Algorithm

Active Clustering is works on the basis of grouping similar nodes and increase the performance of the algorithm the process of grouping is based on the concept of match maker node. Match maker node forms connection between its neighbours which is like as the initial node .Then the matchmaker node disconnects the connection between itself and the initial node. The above set of processes is repeating again and again. Active Clustering is a clustering based algorithm which introduces the concept of clustering in cloud computing. The performance of an algorithm can be enhanced by making a cluster of nodes. Each cluster can be assumed as a group. The principle behind active clustering is to group similar nodes together and then work on these groups. The process of creating a cluster

Volume: 11 Issue: 01 Pages: 4177-4183(2019) ISSN: 0975-0290

revolves around the concept of match maker node. In this process, first node selects a neighbor node called the matchmaker node which is of a different type. This matchmaker node makes connection with its neighbor which is of same type as the initial node. Finally the matchmaker node gets detached. This process is followed iteratively. [10]

Advantage

- The performance of the system is enhanced with high availability of resources, thereby increasing the throughput. This increase in throughput is due to the efficient utilization of resources.

Disadvantage

- Degrades as system diversity increases.

7.) Throttled Algorithm

The sequence of steps:
Step 1. Throttled Load Balancer execution load balancing by update and maintain an index table contains the status information (available '0' or not available '1') of all VMs. At start, all VM at the status is available '0'.
Step 2. Data Center Controller received a new request.
Step 3: Data Center Controller query to Throttled Load Balancer for the new task.
Step 4: Throttled Load Balancer will be checked VM on the top table, determined the first VM is available.
If found VM:

- Throttled Load Balancer sends the ID of VM to Data Center Controller.
- The Data Center Controller sends a request to the VM specified by that ID.
- Data Center Controller notifies the Throttled LoadBalancer a new allocation.
- Throttled LoadBalancer updates the index and waits for new requests from the DataCenter Controller.

If not found VM:

- Throttled Load Balancer will return a value of -1 to the Data Center Controller.
- The Data Center Controller arranges the request.

Step 5: As for the VM, after processing the request and the Data Center Controller receives a response, it will notify to Throttled Load Balancer is stopped.
Step 6: If there are multiple requests, the Data Center Controller repeats Step 3 with the next index and the process is repeated until the index table size is empty.
This algorithm optimizes the response time than the Round Robin algorithm. But the limitation is to detect the VM is ready '0' with the index table size out.[3]

Advantages

- Easy to implement.
- Suitable for small and static system.
- The only single scheduler is required.
- No queues are maintained at the VM level. Only one queue is maintained at scheduler level.[1]

Disadvantages

- Centralized in nature.
- Waiting time is generally large.
- Not suitable for dynamically changing distributed environment
- Does not consider the size of the task and the capacity of VM while task VM mapping.

8. Hierarchical Load Balancing

In order to federate the LB network service, we adopted the notion of a hierarchical LBaaS architecture composed of two layers: a distributor layer responsible for receiving the client traffic and spreading it evenly and consistently over an elastically growing set of load balancing nodes, which are represented by the second layer of hierarchy in Figure 1. The role of the Distributor is two-fold; it must distribute traffic between cloud regions and it must distribute traffic between load-balancer instances within a cloud. The performance goal of the former is dominated by latency requirements; namely, the Distributor should send traffic to the cloud"nearest" to the client. This kind of Distribution, also known as Global Load Balancing, is common to Content Distribution Networks (CDNs) and is typically implemented through DNS.

One of the challenges of Global-Load-Balancing for CDNs is that it must take into account the total available capacity in each region and react fast enough to avoid over- loading any one region. In a federated cloud setting, the federated management has the ability to scale each of the regions as needed to match the demand. In this architecture the Distributor thus performs initial traffic steering between a set of load balancing nodes that perform advanced LB VNFs (Virtual Network Functions), such as application-layer traffic steering and SSL termination.[15]
The second layer in LBaaS hierarchy is designed to elastically grow and shrink similarly to the application servers pool. The VNFs at this layer, i.e., the individual load balancer nodes, are placed to have optimal reachability to the application server nodes and can be located in different data centers of the federated cloud environment interconnected by a federated network. This realization of the LBaaS is based on OpenStack, the architecture is platform independent and can be used to implement federated LBaaS in any federated cloud environment.

Following are the principles of hierarchical load balancing algorithms:
1. They are distributed and heterogeneous.
2. Reduced communication delay.
3. Load index based on CPU Utilization, Queue length and Communication delay acts as a decision factor for scheduling the tasks.
4. The various criteria associated are total execution time, number of tasks and the number of nodes.

Volume: 11 Issue: 01 Pages: 4177-4183(2019) ISSN: 0975-0290

Advantages :

- The Hierarchical structure provides a better performance when compared to other load balancing. The information flow is eased through tree and message traffic is well defined.
- Easy management of Cluster and Node supports heterogeneity and scalability of clouds.

Disadvantages:

- Complexity occurs due to selection and maintenance of cluster heads.
- Slow response time.
- Difficult to manage the workload of the people.
- Unfamiliarity with application or solution domain area.

9. Weighted Round Robin

It is similar to the round robin algorithm in such a way that the distributing manner remained cyclical by which client requests are provided to the nodes. In the configuration of the load balancer, user provides "weights" parameter to each and every node. The node with the higher weight has the ability to handle more no of client requests. For example, If server 1 is more capable and is having more specifications than server 2.Server 1's ability is 4x more than that of server 2's capacity, then user can assign a weight of 4 to server 1 and weight of 1 to the server 2. So when user's request come, the first 4 requests will be assigned to server 1 and the 5th request to server 2. If more requests comes, the similar procedure will be followed in such a way that, the 6th, 7th,8th, 9th request will go to server1, and the 10th request to Server 2, and so on.[2]

Weighted round-robin is an extension of round-robin strategy where servers are given different weight numbers according to their capability. The servers receive service requests in the order of higher weighted server to lower weighted server. As the capability of the servers is considered in this strategy, it will provide better performance than the round-robin strategy.[2]

Advantages:

- Takes care of the capacity of the servers in the group.
- Improvement over round robin.
- Useful for nodes of different capacities.

Disadvantages:

- Does not considered the advanced load balancing requirements such as processing time for each individual request.
- Not useful if tasks have different execution time.

10. Weighted Least Connection

This algorithm is based on the Least Connection method. Like in the Weighted Round Robin method each server is given a numerical value. The load balancer uses this when allocating requests to servers. If two servers have the same number of active connections then the server with the higher weighting will be allocated the new request.

The Weighted Least Connections algorithm does to Least Connections what Weighted Round Robin does to Round Robin. That is, it introduces a "weight" component based on the respective capacities of each server. Just like in the Weighted Round Robin, you'll have to specify each server's "weight" beforehand.

A load balancer that implements the Weighted Least Connections algorithm now takes into consideration two things: the weights/capacities of each server AND the current number of clients currently connected to each server.[2]

Advantages:

- Weighted Least Connections methods work best in environments where the servers have differing capacities.

Disadvantages:

- The Weighted Least Connections methods do not include idle connections in the calculations when selecting a pool member or node. The Weighted Least Connections methods use only active connections in their calculations.

11. Source IP Hash

Source IP Hash load balancing uses an algorithm that takes the source and destination IP address of the client and server and combines them to generate a unique hash key. This key is used to allocate the client to a particular server. As the key can be regenerated if the session is broken this method of load balancing can ensure that the client request is directed to the same server that it was using previously. This is useful if it's important that a client should connect to a session that is still active after a disconnection. For example, to retain items in a shopping cart between sessions.

Advantages:

- A more even distribution of the load compared to Route Based on Originating Virtual Port and Route Based on Source MAC Hash, as the virtual switch calculates the uplink for every packet.
- A potentially higher throughput for virtual machines that communicate with multiple IP addresses.

Disadvantages:

- Highest resource consumption compared to the other load balancing algorithms.
- The virtual switch is not aware of the actual load of the uplinks.
- Requires changes on the physical network.
- Complex to troubleshoot.

Volume: 11 Issue: 01 Pages: 4177-4183(2019) ISSN: 0975-0290

Comparison Table:

Parameters	Min-Min	Max-Min	Round Robin	Honey Bee	Ant-Colony	Biased Random Sampling	Throttled	Active Clustering
Static Environment	Yes	Yes	Yes	No	No	No	No	No
Dynamic Environment	No	No	No	Yes	Yes	Yes	Yes	Yes
Centeralized Balancing	Yes	Yes	Yes	No	No	No	Yes	No
Distributed Balancing	No	No	No	Yes	Yes	Yes	Yes	Yes
Reliability	Less	Less	Less	Less	Low	Yes	Yes	Yes
Throughput	Yes	Yes	Yes	No	No	Yes	No	Yes
Fault Tolerance	No	No	No	No	No	No	Yes	No
Response Time	Yes	Yes	Yes	No	No	No	Yes	No
Resource Utilization	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Migration Time	No	No	No	No	Yes	No	Yes	Yes
Overhead	Yes	Yes	Yes	No	No	Yes	No	Yes
Scalability	No	No	Yes	No	Yes	No	Yes	No
Performance	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Power Saving	No	No	No	No	No	No	No	No

Conclusion:

Load balancing is one of the main challenges in cloud computing. In this paper, we discussed about various techniques present in the static load balancing and dynamic load balancing algorithms. We also discussed the characterstics, advantages and disadvantages of different type of load balancing algorithm.

References:

1. Nguyen Xuan Phi, Cao Trung Tin, Luu Nguyen Ky Thu and Tran Cong Hung, '"Proposed Load Balancing Algorithm to Reduce Response Time and Processing Time On Cloud Computing", International Journal of Computer Networks & Communications (IJCNC), 2018

2. Gurasis Singh, Kamalpreet Kaur, "An Improved Weighted Least Connection Scheduling Algorithm for Load Balancing in Web Cluster Systems", International Research Journal of Engineering and Technology (IRJET), p-ISSN: 2395-0072, 2018.

3. Violetta N. Volkova, Liudmila V. Chernenkaya, Elena N. Desyatirikova, Moussa Hajali, Almothana Khodar, Alkaadi Osama, "Load Balancing in Cloud Computing" IEEE, 2018.

4. Devika Singh1, Ashish Vashisht, "An Improved Regulated Load Balancing Approach for Cloud Computing Environment", International Journal of Research in Advent Technology, E-ISSN: 2321-9637, 2018,

5. Sambit Kumar Mishra, Md Akram Khan, Bibhudatta Sahoo, Deepak Puthal, Mohammad S. Obaidat and KF Hsiao, "Time efficient dynamic threshold-based load balancing technique for Cloud Computing" IEEE International Conference on Computer, Information and Telecommunication Systems (CITS) , Dalian, China, 21-23 July 2017.

6. Desyatirikova E. N., "Kuripta O. V. Quality management in IT service management based on statistical aggregation and decomposition approach","Quality Management,Transport and Information Security, Information Technologies" (IT&QM&IS) , 2017, pp. 500-505. DOI: 10.1109/ITMQIS.2017.8085871.

7. Sobhan Omranian-Khorasani and Mahmoud Naghibzadeh, "Deadline Constrained Load Balancing Level Based Workflow Scheduling for Cost Optimization", IEEE International Conference on Computational Intelligence and Applications, Beijing, China 8-11 Sept 2017.

8. Atyaf Dhari, Khaldun I. Arif , "An Efcient Load Balancing Scheme for Cloud Computing",Indian Journal of Science and Technology,Vol 10(11), DOI:10.17485/ijst/v10i11/110107/2017.

Volume: 11 Issue: 01 Pages: 4177-4183(2019) ISSN: 0975-0290

9. Simar P.S. Anju S. and Rajesh K.," Analysis of load balancing algorithms using cloud analyst, International Journal of Grid and Distributed Computing", vol. 9, No. 9, pp.11-24,2016.

10. D. Saranya, L. Sankara Maheswari, "Load Balancing Algorithms in Cloud Computing: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Issue 7, 2015.

11. 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, P-ISSN 2347 – 5161,2015.

12. Monika Kushwaha, Saurabh Gupta, "Various Schemes of Load Balancing in Distributed Systems- A Review", International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN 2278 – 0882, Issue 7, J2015.

13. Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta and Kuwar Pratap Singh, Nitin and Ravi Rastogi,"LoadBalancingofNodesinCloudUsingAnt colonyOptimization",International Conference on Modelling and Simulation, 2012.

14. Calheiros R.N. ,"CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, Eprint: Australia, pp.9–17,2009.

15. Anna Levina, Dean Lorenza, Giovanni Merlinob,Alfonso Panarellob, Antonio Puliafitob, Giuseppe Tricomib,"Hierarchical Load Balancing as a service for federated Cloud networks", Computer communications (2018), doi: 10.1016/j.comcom.2018.07.031.