Journal of Engineering and Applied Sciences 12 (Special Issue 9): 8579-8585, 2017
ISSN: 1816-949X

A Review Paper on Various Load Balancing Algorithms in Cloud Computing

M. Archana and Mallikarjuna Shastry
School of Computing and Information Technology, REVA University, Kamataka, India

Abstract: Cloud computing has seen remarkable growth and adoption in recent times based on on-demand resource usage principles. Two critical challenges in the usage of cloud computing are cloud security and performance stability. Towards performance stability, load balancing plays a vital role in cloud computing to enhance throughput, optimize resource use and reduce response time. Important criteria to be considered while selecting a load balancing algorithm for cloud is the ability of the algorithm to address distributed network, dynamic environment and self regulation. With the proliferation of data centers, additional challenge in cloud computing paradigm is having energy efficient load balancing in a dynamic and scalable environment with an undefined number of heterogeneous resources. In this study, we discuss the topic of load balancing and survey the different algorithms that are proposed for distributing the load among the nodes and also the parameters that are taken into account for calculating the optimal algorithm to balance the load.

Key words: Cloud computing, load balancing, cloud security, performance stability, optimal, algorithm, dynamic environment

INTRODUCTION

Cloud computing provides the availability of IT resources which are at different parts of the world to users who wants to access those resources from their work place in the form as a service through an optimized and reliable service provider maintaining convenience and ubiquity. The concept of pay-as-you-go applied by cloud providers fits in both economic and technological trends of different organizations. Particularly, the use of virtualization facilitates the sharing of resources and the administration of the whole cloud system. Cloud computing has changed the IT companies way of trading and designing their products. Cloud computing has minimized the incurring premium costs for products with its pay-as-you-go model which leads to increased traffic in the IT services making load balancing a central point of research. To balance the load among multiple resources in a cloud there are several algorithms proposed but till now no algorithms was able to balance the load in a cloud without performance degrading.

In summary, we note that cloud is a distributed system which shares thousands of computing resources. The cloud uses internet links to connect users to providers. The distribution of workloads between different nodes within the cloud network is a crucial step in the process of optimizing the overall workloads. Hence, the importance of load balancing to allocate efficiently the various available resources including cloud network links, central processing units, disk drives or other resources.

Particularly, using a unique hardware load balancer for central administration of hundreds of processors proves costly in terms of time and money. So, to develop an optimized load balancing algorithm one must try to create an environment in which total load of a system can be reassigned to multiple components that work collectively in that system, so that, there will be no overload and under loaded nodes are effectively utilized which reduces the overall response time of the system there by improving the speed, security and reliability. While developing a load balancing algorithm one must consider some important things like estimating the load appropriately, monitoring the performance and stability of the system while performing a task and selection of proper nodes.

MATERIALS AND METHODS

Cloud computing and load balancing architecture: The deployment and service model (Ramya *et al*., 2014) in cloud computing can be illustrated using framework in Fig. 1.

Software as a Service (SaaS): SaaS (Ramya *et al*., 2014) is becoming more prevalent delivery model as underlying technologies that support web services and Service Oriented Architecture (SOA) mature and new developmental approaches such as AJAX become popular. Meanwhile, broadband service has become increasingly available to back up user access from more nations around the globe.

Corresponding Author: M. Archana, School of Computing and Information Technology, REVA University, Kamataka, India

Platform as a Service (PaaS): PaaS (Ramya *et al*., 2014) serves the platform for users who is evolving the software. The service delivery model allows the customer to rent virtualized servers and associated services for moving existing applications or producing and trying new ones.

PaaS offerings may also include facilities for application plan, application progress, testing and deployment as well as services such as team collaboration, web serviceincorporation and marshalling, database integration, security, scalability, storage, persistence, state administration, application versioning, application instrumentation and developer community facilitation.

Infrastructure as a Service (IaaS): IaaS (Ramya *et al*., 2014) serves machines, storage and network resources that developers can manage by installing their own operating system, applications and support resources. The service provider owns the equipment and is responsible for housing, extending and preserving it. The client typically pays on a per usage basis. The definition includes such offerings as practical server space, net connections, bandwidth, IP addresses and load balancers. Iaascan be used by enterprise customers to produce cost efficient and easily scalable IT solutions where the difficulty and expenses of managing the underlying.

The objectives of our study include the realization of comparative analysis of the different load balancing algorithms within the cloud in order to propose a more efficient solution. In this study, some of the optimistic algorithms are surveyed which had shown some improvement in load balancing and increased the level of performance.

Figure 2 shows the load balancing structural design. There are two reasonable units in the load balancing architecture (Ramya *et al*., 2014) namely, load balancers and a controller service. The load balancer are resources that monitor traffic and handle requests from net. The controller service supervises the load balancer, provides information on the capacity requirements and ensures load balancer performs accurately.

Characteristics of load balancing algorithms: While aiming for better load balancing and fulfillment of requests, the following goals need to be achieved.

Scalability: The load balancing algorithm must provide scalability in terms of addition of new resources to address the everyday increasing demand of services with

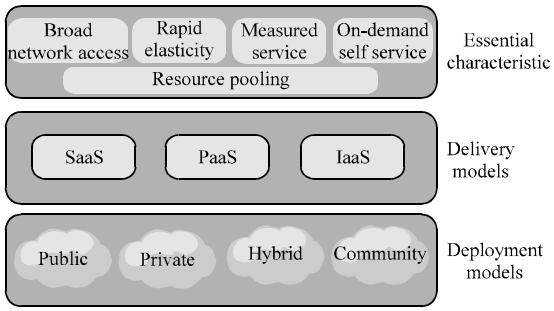

Fig. 1: Cloud computing framework

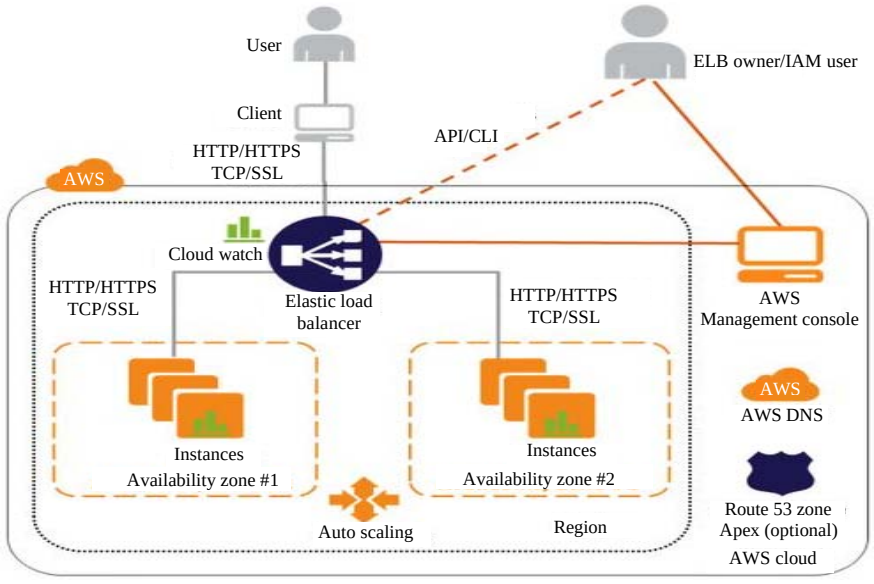

Fig. 2: Load balancing architecture

a greatly increasing number of users. This also demands flexibility in accommodating the augmentation or withdrawal of resources dynamically.

Better response time: The load balancing must beimplemented and executed well enough to provide the best possible response to the user.

Cost effectiveness: A good load balancing algorithm must aim for better overall system performance with the cost being quite reasonable.

Prioritization: The tasks must be prioritized so that the critical tasks do not have to face the problem of starvation or if addressed, the problem of late response.

Fair node utilization: The serving nodes must be utilized efficiently, so that, no single node is overwhelmed, leaving certain others totally free or lightly loaded.

Challenges in load balancing algorithms: Here, we discuss the challenges to be addressed when attempting to propose an optimal solution to the issue of load balancing in cloud computing (Kanakala *et al*., 2015). These challenges are summarized in the following points.

Distribution of cloud nodes: There are many algorithms being proposed for load balancing in cloud computing. Among them some algorithms might produce efficient results with small networks or a network with closely located nodes. Such algorithms are not suitable for large networks because those algorithms cannot produce the same efficient results when applied to larger networks. There are many reasons that affect the efficiency in larger networks like speed of the network, distance between the clients and server nodes and also the distance between all the nodes in the network, so, while developing a load balancing algorithm one should try for better results in spatially distributed nodes balancing the load effectively reducing network delays. Results in spatially distributed nodes balancing the load effectively reducing network.

Algorithm complexity: Load balancing algorithms are preferred to be less complex in terms of implementation and operations. The higher implementation complexity would lead to a more complex process which could cause some negative performance issues. Furthermore, when the algorithms require more information and higher communication for monitoring and control, delays would cause more problems and the efficiency will drop. Therefore, load balancing algorithms must be designed in the simplest possible forms.

Single point failure: Definitely centralized load balancing algorithms (having one controller) can provide efficient results while balancing the load than the distributed algorithms. But in centralized load balancing algorithmswhen the controller failsthe wholesystem will be halted in such cases there will be a huge loss for both client and service provider. So, the load balancing algorithms must be designed in a decentralized anddistributed fashion so that when a node acting as a controller fails the system will not halt (Usurelu *et al*., 2015). In such cases the control will be given to other nodes and they will act as controllers of the system.

Migration time: It is the amount of time for a process to be transferred from one system node to another node for execution. For better performance of the system this time should be always less.

Security: Security is one of the problems that cloud computing has as its top priority (Subashini and Kavitha, 2012). The cloud is always vulnerable in one or the other way to security attacks like DDOS attacks, etc. While balancing the load there are many operations that take place like VM migration, etc. at that time there is a high probability of security attacks. So, an efficient load balancing algorithm (Randles *et al*., 2010) must be strong enough to reduce the security attacks but should not be vulnerable.

Resource utilization: It is the parameter which gives the information of which extent the resource is utilized. For efficient load balancing in system, optimum resource should be utilized, to help in both performance and stability vectors.

Associated overhead: It describes the amount of overhead during the implementation of the load balancing algorithm. It is a composition of movement of tasks, inter process communication and inter processor. For load balancing technique to work properly, minimum overhead should be there (Fig. 3).

Energy management: A load balancing algorithm should be designed in a way such that the operational cost and the energy consumption of the algorithm must be low. Increase in the energy consumption is one of the main problems that cloud computing is facing today. Though mechanisms like energy efficient hardware architectures which slow down the processor speed and turn off machines are deployed, it is still not sufficient. So, to achieve better results in energy management a load balancing algorithm should be designed by following energy aware job scheduling methodology.

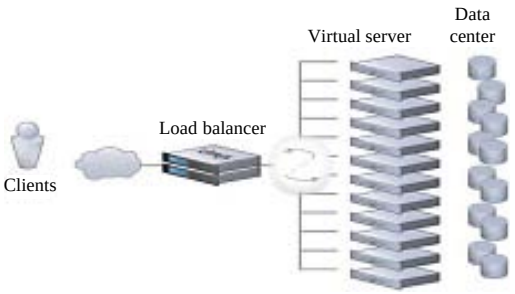

Fig. 3: Static load balancing algorithm

Response time: In distributed system, it is the time taken by a particular load balancing technique to respond. This time should be minimized for better performance.

Performance: It is the overall efficiency of the system. If all the parameters are improved then the overall system performance can be improved.

Types of load balancing algorithm: There are two main types of load balancing algorithms.

Static load balancing algorithm: In static algorithm (Singh *et al*., 2016; Ghutke and Shrawankar, 2014), the traffic is divided evenly among the servers. This algorithm requires a prior knowledge of system resources, so that, the decision of shifting of the load does not depend on the current state of system. Static algorithm is proper in the system which has low variation in load.

Advantages: They will not cause any run-time overhead. They are attractive for parallel programs for which the execution times of processes and their communication requirements can be predicted. In some situations, static load balancing is the only compatible choice.

Disadvantages: It is not satisfactory for parallel programs that are of the dynamic or unpredictable kind.

Dynamic load balancing algorithm: In dynamic algorithm (Wu *et al*., 2012), the lightest server in the whole network or system is searched and preferred for balancing a load. For this real time communication with network is needed which can increase the traffic in the system. Here current state of the system is used to make decisions to manage the load (Ren *et al*., 2011).

Advantages: They incur non negligible run-time overhead. They are aiming for a limited balanced state between the two extremes representing a certain tradeoff among balancing quality and run-time overhead.

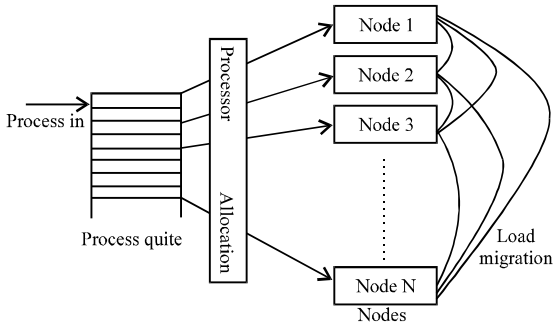

Fig. 4: Dynamic load balancing algorithm

Disadvantages: The design and analysis of dynamic's load balancing algorithms is a complex process (Fig. 4).

RESULTS AND DISCUSSION

Comparative analysis of algorithms: In this study, the different algorithm are discus and compare these algorithms based on challengessuch as spatial distribution of cloud node, algorithm complexity, storage of data and point of failure in algorithm.

Weighted round-robin load balancing algorithm: Weighted Round Robin Algorithm (Singh *et al*., 2016) is the modified version of the round robin algorithm. In this algorithm, a weight is assigned to each and every server or node and distribution of the job is depends on the values of the weights. Larger capacities processors have the larger value of weight. In a situation where all weights become equal, servers will receive balanced traffic.

Pros of round robin load balancing algorithm: Modified version of round robin. Work well for nodes of different capacities.

Cons of weighted round robin load balancing algorithm: Not work well if nodes have different execution time.

Min-min load balancing algorithm: Min-min load balancing algorithm (Singh *et al*., 2016; Chen *et al*., 2013) begins by finding the minimum completion time for all tasks. Then among these minimum times, the minimum value is selected which is the minimum time amongst all tasks on any resource. According to that minimum time, the task is then scheduled on the corresponding machine. The execution time for all other tasks is updated on that machine and that task is removed from the list. This procedure is followed until all the tasks are assigned the resource. In scenarios where the number of small tasks is more than the number of large tasks this algorithm achieves better performance. However, this approach can lead to starvation.

Pros of min-min load balancing algorithm: Reliable tasks assignment to the nodes.

Cons ofmin-min load balancing algorithm: Slower than other algorithms because work must pass through three layers to be processed.

Max-min load balancing algorithm: Max-min load balancing algorithm (Elzeki *et al*., 2012) is similar to the min-min algorithm except the following: after finding out the minimum execution times, the maximum value is selected which is the maximum time amongst all tasks on the resources. Then according to the maximum time, the task is scheduled on the corresponding machine. The execution time for all other tasks is updated on that machine and the assigned task is removed from the list of tasks that are to be assigned to the machines. Since, the requirements are known beforehand this algorithm is expected to perform well.

Pros of max-min load balancing algorithm: High efficiency and performance.

Cons of max-min load balancing algorithm: Waiting of high time consuming tasks is very long.

Ant colony algorithm: In ant colony algorithm (Ragmani *et al*., 2016; Zhang and Zhang, 2010), this algorithm is developed to find out the best path between the food and ant colony this is done to equally spread the load between all nodes. The local node is referred as the head node. When the load is tagged to the head node, ant with loaded node start moving forward to find out the node either node is overloaded or not. Whenever ant find out any loaded node it move further and if find out any overloaded node it move in reverse direction to displace the previously found node. At that time when all task completes then the result is modified or updated to build a final result report.

Pros of ant colony optimization: All the information about nodes collected by ant is quite fast.

Cons of ant colony optimization:

- Over head on network is very high
- Number of ants does not doubtlessly define
- Dynamic load balancing algorithm

Throttled load balancing algorithm: Throttled load balancing algorithm (Ramya *et al*., 2014) absolutely established on Virtual Machine (VM). In this algorithm, the throttled load balancer search the acceptable VM to execute the client operation this is done only when client requested the load balancer to search the acceptable VM. The searching of acceptable VM become an issue of this algorithm, the issue is delay in operation. The throttled load balancer keep up the state (available/busy) along an index table of virtual machines.

Pros of throttled load balancing algorithm: Very efficient and performance is really good.

Cons of throttled load balancing algorithm: Delay in operation due to long search to find acceptable virtual machines.

Neighbour aware random sampling algorithm: In neighbour aware random sampling (Ariharan and Manakattu, 2015) when the load balancer of a node receives a job request, it sends a request token with the duration of resource request and load of current node to a randomly selected neighbour. The node that receives the request token adds its own load and forwards to a randomly selected neighbour until the number of nodes covered by the request token is equal to the walk threshold of the network, the last node to receive the request token selects the least loaded node from the request token and allocates the job to thatnode. The last node in the walk sends allotment token to the node which is selected for the job. Upon receiving the allotment token, the allotted node can receive the job details such as inputs of the job, executable and other particulars from the node which initiated the random walk.

Pros of neighbour aware random sampling algorithm: Communication delay is low as compare to other algorithms, not centralized.

Spider mesh overlay: The spider mesh overlay (Usurelu *et al*., 2015) proposed is similar to that of a spider web: the overlay is composed of chains extending from a center point and rings connecting these chains. Also, chains can have varying lengths thus implying that rings can be complete or discontinuous. The chains and rings represent virtual communication paths while their intersection represents computational nodes. The nodes on each chain are identicalbut two chains are not necessarily identical. The central pointof the network is represented by the task distributor. The distributor's purpose is to distribute tasks to chains, according to the task resource needs. A task matches a node based on two conditions firstly, the node must have the necessary resources in order to carry out the task and secondly, adding the task must not exceed the nodes available resource capacities.

Pros of spider mesh overlay: Effeciently used for load balancing non preemptive tasks. Round robin approach or clustering approach can be used for load balancer.

Table 1: Analysis of algorithm

Load balancing algorithm	Static/Dynamic	Performance	Throughput	Fault tolerance	Response time	Over head	Migration time
Weighted round robin	Static	High	Low	No	High	High	High
LB min-min	Static	Low	Moderate	No	Low	High	Moderate
LB max-min	Static	High	High	No	Low	Moderate	Moderate
Ant colony optimizatiom	Dynamic	Low	Low	No	High	High	Low
Throttled load balancing algorithm	Dynamic	High	High	Yes	High	Low	Moderate
Neighbour aware random sampling algorithm	Dynamic	High	High	Yes	High	Low	Low
Spider mesh overlay	Dynamic	High	High	Yes	High	Low	low

Cons of spider mesh overlay: Routing policies are not efficient. So, current routing policy should be enhanced or new routing policies need to be added.

Analysis of algorithm based on the following parameters: Now, we analyse all the discussed algorithm with the parameter such as: performance, complexity, scalability, fault tolerance, response time, throughput and many more. (Table 1).

CONCLUSION

In this study, the significance of load balancing in cloud computing and various challenges associated with balancing load in a cloud computing network are discussed. Additionally the classification and characteristics of various load balancing algorithms were surveyed and presented, each algorithm discussed with its advantages and disadvantages. Given the increasing adoption of cloud based services, it is becoming increasingly challenging to meet the requirements of scalability, security and responsiveness through conventional models in the existing algorithms.

RECOMMENDATION

Future direction would be to look into new adaptive models and metrics to be able to develop an efficient load balancing algorithm meeting performance requirements.

REFERENCES

Ariharan, V. and S.S. Manakattu, 2015. Neighbour Aware Random Sampling (NARS) algorithm for load balancing in cloud computing. Proceedings of the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), March 5-7, 2015, IEEE, Coimbatore, India, ISBN:978-1-4799-6084-2, pp: 1-5.

Chen, H., F. Wang, N. Helian and G. Akanmu, 2013. User-priority guided min-min scheduling algorithm for load balancing in cloud computing. Proceedings of the National Conference on Parallel Computing Technologies, February 21-23, 2013, Bangalore, India, pp: 1-8.

Elzeki, O.M., M.Z. Reshad and M.A. Elsoud, 2012. Improved max-min algorithm in cloud computing. Int.J. Comput. Applic., 50: 22-27.

Ghutke, B. and U. Shrawankar, 2014. Pros and cons of load balancing algorithms for cloud computing. Proceedings of the 2014 International Conference on Information Systems and Computer Networks (ISCON), March 1-2, 2014, IEEE, Mathura, India, ISBN:978-1-4799-2981-8, pp: 123-127.

Kanakala, V.R., V.K. Reddy and K. Karthik, 2015. Performance analysis of load balancing techniques in cloud computing environment. Proceedings of the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), March 5-7, 2015, IEEE, Coimbatore, India, ISBN:978-1-4799-6084-2, pp: 1-6.

Krishna Reddy, V., B. Thirumala Rao, L.S.S. Reddy and P. Sai Kiran, 2011. Research issues in cloud computing. Global J. Comput. Sci. Technol., 11: 59-64.

Ragmani, A., E.A. Omri, N. Abghour, K. Moussaid and M. Rida, 2016. A performed load balancing algorithm for public cloud computing using ant colony optimization. Proceedings of the 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), May 24-26, 2016, IEEE, Marrakech, Morocco, ISBN:978-1-4673-8894-8, pp: 221-228.

Ramya, R., M. Kriushanth and L. Arockiam, 2014. A State-of-Art Load balancing algorithms in cloud computing. Intl. J. Comput. Appl., 95: 10-14.

Randles, M., D. Lamb and B.A. Taleb, 2010. A comparative study into distributed load balancing algorithms for cloud computing. Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), April 20-23, 2010, IEEE, Perth, Australia, ISBN: 978-1-4244-6701-3, pp: 551-556.

Ren, X., R. Lin and H. Zou, 2011. A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast. Proceedings of the IEEE International Conference on Cloud Computing and Intelligent Systems, September 15-17, 2011, Beijing, China, pp: 220-224.

Singh, P., P. Baaga and S. Gupta, 2016. Assorted load balancing algorithms in cloud computing: A survey. Intl. J. Comput. Appl., 143: 34-40.

Subashini, S. and V. Kavitha, 2012. An adaptive security framework delivered as a service for cloud environment. J. Eng. Appl. Sci., 7: 468-482.

Usurelu, C.C., M.C. Nita, R. Istrate, F. Pop and N. Tapus, 2015. Spider mesh overlay for task load balancing in cloud computing. Proceedings of the 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), September 3-5, 2015, IEEE, Cluj-Napoca, Romania, ISBN: 978-1-4673-8200-7, pp: 433-440.

Wu, T.Y., W.T. Lee, Y.S. Lin, Y.S. Lin, H.L. Chan and J.S. Huang, 2012. Dynamic load balancing mechanism based on cloud storage. Proceedings of the Computing, Communications and Applications Conference, January 11-13, 2012, Hong Kong, pp: 102-106.

Zhang, Z. and X. Zhang, 2010. A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. Proceedings of the 2nd International Conference on Industrial Mechatronics and Automation, Volume 2, May 30-31, 2010, Wuhan, China, pp: 240-243.