An Analysis of Load Balancing in Cloud Computing

Suresh M.
PG Scholar
SNS College of Technology,
Tamilnadu, India

Shafi Ullah Z.
PG Scholar
SNS College of Technology,
Tamilnadu, India

Santhosh Kumar B.
Assistant Professor
SNS College of Technology,
Tamil Nadu, India

Abstract:

Cloud computing is becoming a key technology for online allotment of computing resources and online storage of user's data in a lower cost, where computing resources are available all the time, over the internet with pay per use concept. Cloud computing is a term, which involves virtualization, distributed computing, utility computing, networking, software and web services .As advancements of various areas of technology increases, different types of issues have been introduced in cloud. In this paper we analysis about load balancing of cloud computing with some of the existing load balancing techniques, which are responsible to manage the load when some node of the cloud system is overloaded and others are under loaded. Load balancing ensures that all the processors in the system as well as in the network does approximately the equal amount of work at any instant of time.

Keywords: *Cloud Computing, Load Balancing.*

1 Introduction:

1.1 What is Cloud Computing?

Cloud computing has become very popular in recent years as it offers greater flexibility and availability of computing resources at very low cost [1]. Cloud Computing is a general term used to describe a new class of network based computing that takes place over the Internet, basically a step on from utility computing as shown in Figure 1. The cloud architecture can be split up in three main layers, namely: infrastructure, platform and software .These platforms hide the complexity and details of the underlying infrastructure from users and applications by providing very simple graphical interface or API (Applications Programming Interface).

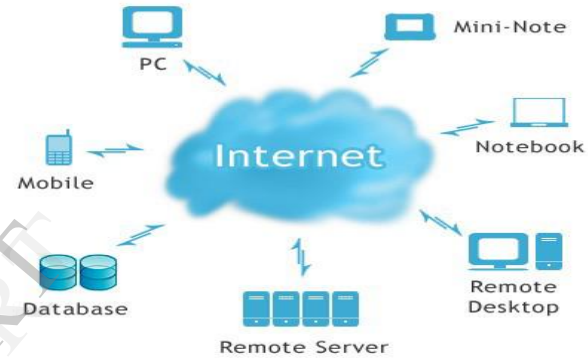

Figure 1: cloud computing

1.2 Cloud Service Models:

NIST [2] defines three main service models for cloud computing:

Software as a service (SaaS):

Instead of buying software and installing it on a local machine, the software-as-a-service model provides users with software-on-demand. When users need to use applications, they use Web interface applications or a providers program, and pay only for the cost of the time they use it. Example: Google Apps (Google Docs) , Sales Force.com .

Platform as a service (PaaS):

Service providers support an environment in which all software and runtime are ready for use; program developers then simply upload their data, such as Web application codes and database data. Example: Google App Engine, Force.com

Infrastructure as a service (IaaS):

Users rent virtual servers for their needs instead of buying real machines and software, and thus save the cost required for both equipment and maintenance. Example: Amazon's Elastic Compute Cloud (EC2), and Simple Storage Service (S3).

1.3 Cloud Deployment Model:

NIST defines cloud deployment models as public, private, community, and hybrid [3]. A cloud system can be operated in one of the following four deployment models:

Private cloud:

These services are managed and provided within the organization. There are less restriction on network bandwidth, fewer security exposures and other legal requirements compared to the public Cloud. Example: HP Data Centers.

Public Cloud:

These services are delivered to the client via the Internet from a third party service provider. Example: Amazon.

Hybrid Cloud:

It is a combination of services provided from public and private Clouds. Example: ERP in Private cloud, Sales & Email on public

Community Cloud:

These services are shared by some organizations and support a specific community that shares concerns .Example: security requirements.

1.4 What is Load Balancing?

Load balancing is dividing the amount of work that a computer has to do between two or more computers so that more work gets done in the same amount of time and, in general, all users get served faster. Load balancing can be implemented with hardware, software, or a combination of both. Load balancing optimizes resource use, maximizes throughput, minimizes response time, and avoids overload [4]. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. When you apply load balancing during runtime, it is called dynamic load balancing this can be realized both in a direct or iterative manner according to the execution node selection.

- In the iterative methods, the final destination node is determined through several iteration steps.
- In the direct methods, the final destination node is selected in one step. These approaches aim to enhance the overall performance of the Cloud and provide the user more satisfying and efficient services.

1.5 Goals of Load Balancing:

Goals of load balancing as discussed by authors of [5] include:

- To improve the performance.
- To maintain the system stability.
- To increase the flexibility of system.
- To have a backup plan in case the system fails even partially

2 Load Balancing Techniques:

2.1 Static Load Balancing:

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more imperfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic [6].

The performance of the virtual machines is determined at the time of job arrival. The master processor assigns the workload to other slave processors according to their performance. The assigned work is thus performed by the slave processors and the result is returned to the master processor.

Static load balancing algorithms are not preemptive and therefore each machine has at least one task assigned for itself. Its aims in minimizing the execution time of the task and limit communication overhead and delays.

The four different types of Static load balancing techniques are Round Robin algorithm, Central Manager algorithm, Threshold algorithm and Randomized algorithm.

2.2 Dynamic Load Balancing:

Dynamic load balancing algorithms, the current state of the system is used to make any decision for load balancing. It allows

For processes to move from an over utilized machine to an under-utilized machine dynamically for faster execution as shown in Figure 2. This means that it allows for process preemption which is not supported in Static load balancing approach. An important advantage of this approach is that its decision for balancing the load is based on the current state of the system which helps in improving the overall performance of the system by migrating the load dynamically [7].

- **Dynamic Load Balancing Policies or Strategies:**

The different policies as described in [8] are as follows:

1. Location Policy:

The policy used by a processor or machine for sharing the task transferred by an over loaded machine is termed as Location policy.

2. Transfer Policy:

The policy used for selecting a task or process from a local machine for transfer to a remote machine is termed as Transfer policy.

3. Selection Policy:

The policy used for identifying the processors or machines that take part in load balancing is termed as Selection Policy.

4. Information Policy:

The policy that is accountable for gathering all the information on which the decision of load balancing is based id referred as Information policy.

5. Load estimation Policy:

The policy which is used for deciding the method for approximating the total work load of a processor or machine is termed as Load estimation policy.

6. Process Transfer Policy:

The policy which is used for deciding the execution of a task that is it is to be done locally or remotely is termed as Process Transfer policy.

7. Priority Assignment Policy:

The policy that is used to assign priority for execution of both local and remote processes and tasks is termed as Priority Assignment Policy.

8. Migration Limiting Policy:

The policy that is used to set a limit on the maximum number of times a task can migrate from one machine to another machine.

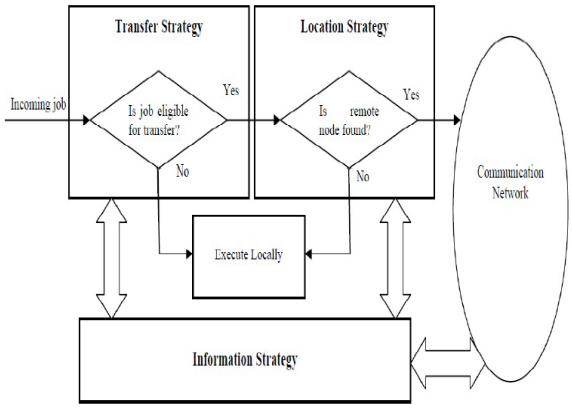

Figure 2: Interaction among components of a dynamic load balancing algorithm

2.3 Ant Colony Optimization:

A cloud is constituted by various nodes which perform computation according to the requests of the clients. As the requests of the clients can be random to the nodes they can vary in quantity and thus the load on each node can also vary. Therefore, every node in a cloud can be unevenly loaded of tasks according to the amount of work requested by the clients. This phenomenon can drastically reduce the working efficiency of the cloud as some nodes which are overloaded will have a higher task completion time compared to the corresponding time taken on an under loaded node in the same cloud. This problem is not only confined only to cloud but is related with every large network like a grid, etc. we propose an efficient algorithm, based on ACO for better distribution of workload among the nodes of a cloud.

The ant uses the basic pheromone updating formula and node selection formula of the ACO to distribute evenly the work loads of nodes in a cloud. For efficient load balancing of work in cloud, tier-wise distribution of nodes is also suggested [9], in this the nodes are distributed in three tier structure such that the work is properly distributed among the nodes. In this hierarchy the 1st level (Top-level) nodes are used for the proper distribution of work among the nodes of 2nd level. Simultaneously the 2nd level distributes the work logically among the 3rd level nodes, which in turn-process their part of work. Thus ,this system ensures the proper distribution of load among all levels.

For building an optimum solution set. We first select a Regional load balancing node (RLBN) [10] is chosen in a CCSP, which will act as a head node. We would be referring to the RLBN as head node in the rest. The selection of head node is not a permanent thing but a new head node can be elected if the previous node stops functioning properly due to some inevitable circumstances. The head node is chosen in such way that it has the most number of neighboring nodes, as this can help our ants to traverse in most possible directions of the network of CCSP.

These ants traverse the width and length of the network in such a way that they know about the location of under loaded or over loaded nodes in the network. These Ants along with their traversal will be updating a pheromone table, which will keep a tab on the resources utilization by each node. We also proposed the movement of ants in two ways similar to the classical ACO, which are as follows:

1) Forward movement-The ants continuously move in the forward direction in the cloud encountering overloaded node or under loaded node.

2) Backward movement-If an ant encounters an overloaded node in its movement when it has previously encountered an under loaded node then it will go backward to the under loaded node to check if the node is still under loaded or not and if it finds it still under loaded then it will redistribute the work to the under loaded node. The vice-versa is also feasible and possible.

The main benefit of this approach lies in its detections of overloaded and under loaded nodes and thereby performing operations based on the identified nodes. This simplistic approach elegantly performs our task of identification of nodes by the ants and tracing its path consequently in search of different types of nodes. The ants continuously update a single result set rather than updating their own result set. In this way, the solution set is gradually built on and continuously improved upon rather than being compiled only once in a while.

2.4 Honey Bee Foraging Behavior:

The honey-bee foraging solution in [11], is investigated as a direct implementation of a natural phenomenon. Then, a distributed, biased random sampling method that maintains individual node loading near a global mean measure is examined. Finally, an algorithm for connecting simile services by local rewiring is assessed as a means of improving load balancing by active system restructuring. In case of load balancing, as the web servers demand increases or decreases, the services are assigned dynamically to regulate the changing demands of the user. The servers are grouped under virtual servers (VS), each VS having its own virtual service queues. Each server processing a request from its queue calculates a profit or reward, which is analogous to the quality that the bees show in their waggle dance.

New population-based search algorithm called the Bees Algorithm (BA). The algorithm mimics the food foraging behavior of swarms of honey bees. In its basic version, the algorithm performs a kind of neighborhood search combined with random search and can be used for both combinatorial optimization and functional optimization. Honey bees have developed the ability to collectively choose between nectar sources by selecting the optimal one: This source provides a maximum ratio of gain compared to costs [12]. The whole decentralized decision process is based on competition among dancing bees, which guide new (naive) bees to their foraging targets. In [13], authors have proposed Load balancing using bees algorithm.

In [14], "On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers", we propose a new honey bee allocation algorithm based on self-organized behavior of foragers in honey bee colonies. Hosting centers then must allocate servers among clients to maximize revenue. The allocation of servers to collect revenue in Internet hosting centers parallels the allocation of foragers to collect nectar in honey bee colonies. A hosting center with a certain number of servers hosting multiple Internet clients is analogous to a honeybee colony with a certain number of bees foraging at multiple sites in the surrounding countryside.
This Insect foraging technique is used in the field of robotics. The main principles of social insect foraging behavior can find an application in a swarm of inexpensive insect-like robots [15].

2.5 Load Balancing Algorithm in VM Cloud:

Load balancing is one of prerequisites to utilize the full resources of parallel and distributed systems. Load balancing mechanisms can be broadly categorized as centralized or decentralized, dynamic or static, and periodic or non-periodic. Physical resources can be split into a number of logical slices called Virtual Machines (VMs). All VM load balancing methods are designed to determine which Virtual Machine assigned to the next cloudlet [4].

Data Center object manages the data center management activities such as VM creation and destruction and does the routing of user requests received from User Bases via the Internet to the VMs .The Data Center Controller [16], uses a Vm Load Balancer to determine which VM should be assigned the next request for processing. Most common Vm Load Balancer is throttled and active monitoring load balancing algorithms.

1. Throttled Load Balancer:

It maintain a record of the state of each virtual machine (busy/ ideal), if a request arrive throttled load balancer send the ID of ideal virtual machine to the data center controller and it allocates the ideal virtual machine.

2. Active Monitoring Load Balancer:

Active VM Load Balancer maintains information about each VMs and the number of request**s** currently allocated to which VM. When a request arrives, it identifies the least loaded VM. If there are more than one, the first identified is selected. Data Center Controller notifies the Active Vm Load Balancer of the new allocation.

The Proposed Load balancing algorithm is divided into three parts. The first phase is the initialization phase. In the first phase, the expected response time of each VM is to be found. In second Phase find the efficient VM, in Last Phase return the ID of efficient VM.

- Efficient algorithms find expected response time of each Virtual machine.

- When a request to allocate a new VM from the Data Center Controller arrives, Algorithms find the most efficient VM for allocation.
- Efficient algorithms return the id of the efficient VM to the Datacenter Controller.
- Datacenter Controller notifies the new allocation
- Updates the allocation table increasing the allocations count for that VM.
- When the VM finishes processing the request and the Data Center Controller receives the Response. Data center controller notifies the efficient algorithm for the VM de-allocation.

We conclude that if we select an efficient virtual machine then it affect the overall performance of the cloud Environment and also decrease the average response time is decrease.

2.6 Cloud Hybrid Load Balancer (CHLB):

The three main components of the CHLB include the RRDNS, the load balancing system and the web system. Each component can be one or groups of virtual machines .To share the efforts of the load balancer and to avoid the main DNS service fail, at least two of the RRDNS VMs include all web IPs information and those RRDNS IP must be registered to the global DNS service provider .The responsibility of the load balancing system is to receive the http requests and then redirect them to the web system. It could be a single VM or a cluster for the high availability purpose. The web system receives the requests from the load balancing system, and transfers the data to the users [17]. If some VMs need to be closed for the system maintenance purpose, the new alternative VMs can be deployed through requests. In the framework, users do not need prepare any hardware machines, network environments and the IT staffs.

Figure 3. Shows the architecture of the proposed cloud hybrid load balancer (CHLB) in the cloud environment. This framework combines with the Web clusters and the load-balancing VMs in the hybrid cloud environment. Each Web cluster includes its own network load balancer (LVS), and the LVSsystems are setup for high availability of a specific Web service. The RRDNS VM is responsible for sequentially arranging the destination IP address of the Web server cluster, and this function can spread the load of the first Web cluster. However, the second RRDNS VM becomes the primary DNS.

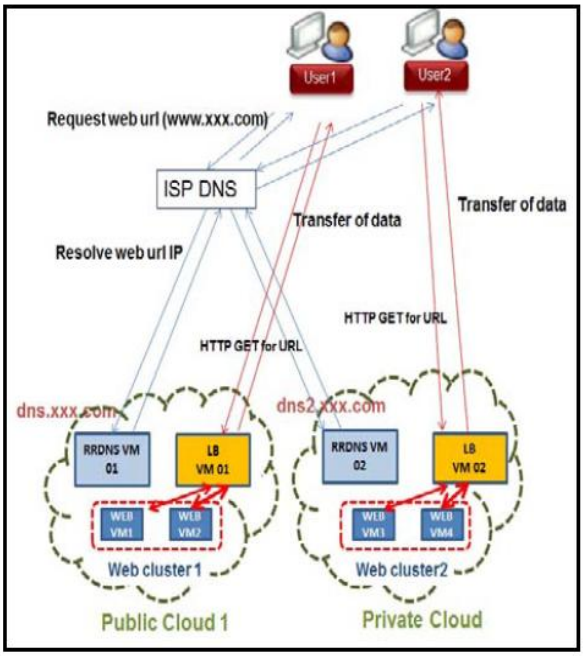

Figure. 3The framework of the CHLB

The cloud hybrid load balancer (CHLB) proposed in this paper is an open-source hybrid load balancing solution for Web clusters. This framework can be applied with other kinds of hypervisors, which can deploy new LVS VM instances, as shown in Fig. 3. The LVS VM is responsible for balancing the loads imposed by the traffic of a group of Web servers. The RRDNS arranges the HTTP requests from the users and transfers the links to the different LVS addresses. Then, the LVS redirects to the real Web server, which returns the HTTP results to the user. The purpose of this research was to develop an open-source solution that can rapidly be reused in the cloud environment, since the costs of these virtual load balancers are much less than those of the customary physical load balancers. Furthermore, the CHLB can decrease the activity of the LVS by means of the RRDNS.

3 Metrics for Load Balancing :

The different qualitative metrics or parameters that. are considered important for load balancing in cloud computing [18] are discussed as follows:

1. ***Throughput:*** The total number of tasks that have completed execution is called throughput. A high throughput is required for better performance of the system.
2. ***Associated Overhead:*** The amount of overhead that is produced by the execution of the load

balancing algorithm. Minimum overhead is expected for successful implementation of the algorithm.

3. ***Fault tolerant:*** It is the ability of the algorithm to perform correctly and uniformly even in conditions of failure at any arbitrary node in the system.

4. ***Migration time:*** The time taken in migration or transfer of a task from one machine to any other machine in the system. This time should be minimum for improving the performance of the system.

5. ***Response time:*** It is the minimum time that a distributed system executing a specific load balancing algorithm takes to respond.

6. ***Resource Utilization:*** It is the degree to which the resources of the system are utilized. A good load balancing algorithm provides maximum resource utilization.

7. ***Scalability:*** It determines the ability of the system to accomplish load balancing algorithm with a restricted number of processors or machines.

8. ***Performance:*** It represents the effectiveness of the system after performing load balancing. If all the above parameters are satisfied optimally then it will highly improve the performance of the system.

4 Conclusion:

Load balancing is one of the main challenges in cloud computing. It is required to distribute the load evenly at every node. A highly congested provider may fall to provide efficient services to its customers. So, with proper load balancing algorithm system service and throughput can be increased. This paper is to focus on one of the major concerns of cloud computing that is Load balancing. The goal of load balancing is to increase client satisfaction and maximize resource utilization and substantially increase the performance of the cloud system thereby reducing the energy consumed and the carbon emission rate.

References:

[1]Rittinghouse, Cloud Computing: Implementation, Management, and Security. 1st edition, CRC Press, 2009, 26. M. Vouk: Cloud Computing—Issues, Research, and Implementations. Proc. 30th Int'l Conf. Information Technology Interfaces, Univ. Computing Centre, Zagreb, Croatia, pp. 235–246, 2008.

[2] Bhaskar Prasad Rimal, Enumi Choi, Ian Lumb, *"A taxonomy and survey of cloud computing systems"*, 5th International Joint Conference in INC, IMS and IDC, 978-0-7695-3769-6/09, 2009, pp 44-51.

[3] Dr. Fang Liu, Jin Tong, Dr. Jian Mao, Knowcean Consulting Inc. "NIST Cloud Computing Reference Architecture" version 1, March 30, 2011.

[4] R. Shinmonski. Windows 2000 & Windows Server 2003." *Clustering and Load balancing*". Emeryville. McGraw-Hill professional publishing, CA, USA (2003), p2, 2003.

[5] David Escalnte and Andrew J. Korty, "*Cloud Services:Policy and Assessment*", EDUCAUSE Review, Vol. 46, July/August 2011.

[6]R. X. T. and X. F. Z.. A Load Balancing Strategy Based on the Combination of Static and Dynamic, in Database Technology and Applications (DBTA), 2010 2nd International Workshop (2010), pp. 1-4.

[7]Meenakshi Sharma, Pankaj Sharma, Dr. Sandeep Sharma,"*Efficient Load Balancing Algorithm in VM Cloud Environment*", IJCST Vol. 3, Iss ue 1, Jan. - March 2012.

[8]Abhijit A Rajguru, S.S. Apte, *"A Comparative Performance Analysis of Load Balancing Algorithms In Distributed Systems Using Qualitative Parameters"*, International Journal of Recent Technology and Engineering, Vol. 1, Issue 3, August 2012.

[9] S.C. Wang, K.Q. Yan, W.P. Liao and S.S. Wang,*" Towards a Load Balancing in a Three-level Cloud Computing Network"*, Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, pp. 108-113,2010.

[10] Kumar Nishant, Pratik Sharma, Vishal Krishna,Chhavi Gupta and Kunwar Pratap Singh "*Load Balancing of Nodes in Cloud Using Ant Colony Optimization* ", proceedings of 14th International Conference on Modelling and Simulation.

[11]M. Randles, A. Taleb-Bendiab, D. Lamb, Scalable self governance using service communities as ambients, in: Proceedings of the IEEE Workshop on Software and Services Maintenance and Management (SSMM 2009) within the 4th IEEE Congress on Services, IEEE SERVICES-I 2009, July 6–10, Los Angeles, CA (to appear), 2009.

[12]T.D. Seeley, Honey bee foragers as sensory units of their colonies, Behavioral Ecology and Sociobiology 34 (1994) 51–62.

[13]A.M. Bernardino, E.M. Bernardino, J.M. Sánchez-Pérez, M.A. Vega-Rodríguez, J.A. Gómez-Pulido, Efficient Load Balancing Using the Bees Algorithm, Trends in Applied Intelligent Systems, Lecture Notes in Computer Science, Springer,Berlin, Heidelberg, 2011.

[14]S. Nakrani, C. Tovey, On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers, Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 12 (3–4 (Sep–Dec)) (2004) 223–240.

[15]J.-L. Deneubourg, S. Goss, R. Beckers, G. Sandini, A. Babloyantz, Self- Organization, Emergent Properties, and Learning, Plenum Press, New York, 1991, p. 267.

[16]Bhathiya Wickremasinghe, Rodrigo N. Calheiros, Rajkumar Buyya,"*CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications"*, 20-23, April 2010, pp. 446-452.

[17]Po-HueiLiang, Jiann-Min Yang," Evaluation of Cloud Hybrid Load Balancer (CHLB)", Feb. 2013, Vol. 3 Iss. 1, PP. 38-42.

[18] Jain Kansal and Inderveer Chana, "Existing Load Balancing Techniques in Cloud Computing: A Systematic Review, Journal of Information Systems and Communication, Vol. 3, Issue 1.