International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.3 (2016), pp.317-332
http://dx.doi.org/10.14257/ijmue.2016.11.3.30

Effort of Load Balancer to Achieve Green Cloud Computing: A Review

G.Siva Shanmugam, N.Ch.S. N. Iyengar

School of Computer Science &Engineering, VIT University, vellore-632014, TN, India
sivashanmugam.g@vit.ac.in , nchsniyr@vit.ac.in

Abstract

In a distributed system, from the starting days onwards distribution of load among servers becomes a serious problem in the commercial Internet. The problem in this scenario is? The entire single application oriented server has to engage the entire amount of traffic and if they went down, all commercial activities come down offline result out of business. Running individually the application server couldn't support start-up budgets. The folks involved in the Web Commercial plan to solve this problem by distributing the load evenly to all the servers running on the web host owned by different agents or organizations; thus new scenario was born named as Load Balancing. In this treatise, we investigate several shapes attenuated by load Balancer and reviewed the algorithms proposed on load balancing. Here we took both static as well as dynamic based algorithms and their performances are formulated by comparison with all other existing scheme. This paper also brings connectivity on green computing with cloud load balancers. By cloud computing we can attain multi tenancy and dynamic resource handling which automatically reduces co2 emission from servers. Without the facility of sharing single resources among thousands of peoples, green computing is not possible. So the nature of cloud load balancer and green computing was illustrated here.

Keywords: *Distributed system, Load Balancer, Load balancing algorithms, Resource provisioning.*

1. Introduction

We know distributed system in which computers are modeled by linear load-dependent latency function [1]. It serves the potential to share different computational resources to other network connected device within in a minimal procedure. A distributed system can be described as collection of several computing resources; these resources are shared and communicated with the users present in the system. By having this feature the user can get the ability to use any type of resource like hardware, software and stored data anywhere in the system. Once the arrival of distributed system from centralized system the commercial agents achieve openness, concurrency, scalability, transparency, fault tolerance and mainly resource sharing. The another functionality extending the life span of distributed system is multiple points of control means, at a time we can run our program in concurrent processes on different processors. The figure 1 shows the model of distributed system.

1.1. Distributing Models

Due to change in application needs distributed system has further extended to grid computing, cloud Computing and go on. Each time these new computational paradigms bring better way of aggregating the resource and serve to other servers. The end users of these paradigms are comes from different applications and maintain different strategy to reach the destination, hence characteristic behavior of

ISSN: 1975-0080 IJMUE

these users may vary with each other. In this environment satisfying their application and maintain their resource are very tedious job and sudden increases on user demand also creates trouble in executing the distributed models. This suppress the performance of distributed system and gives trouble on handling the large number of task and finding computers to allocate new entry task. To bring optimum in distributed system they started to uses load balancer. The same load balancer is practiced in next level computational inventions like grid computing as well as cloud computing, they handle this by modifying the existing algorithms and producing new algorithms. The new paradigms apply artificial intelligent system on load balancer and achieve the load distribution to the present servers. Load balancer deal with the management of distributed system is to have a unique decision maker that will lead the system to its optimum [2].

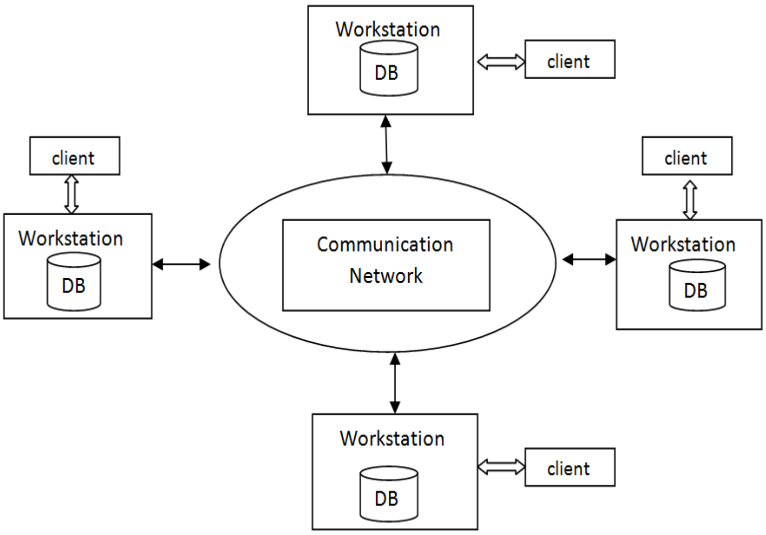


Figure 1.Model of Distributed System

1.2. Grid Computing

The Grid computing is based on distributed architecture were all the computers inside the models are interconnected which provide massive computational power on solving a task. Under this model all the servers run their individual task and they are loosely connected with communicated network, those results were aggregated by scheduling algorithms. The model view of grid is shown in fig 1.2.

The concept behind the grid is difficult when comparing with normal distributed system. The distributed model suits only for narrow bounder applications, dependent modules or subset related task are not executed under this model were as grid support highly connected applications. The application of grid promises recovery of robust and redundancy failures of system. Grid achieved this by having controls on member nodes are loosely connected and it provides the facility to exchange and handle the jobs from any geographically scattered locations. Inside the grid, all the member nodes are act like a server. So any node can be access the request initiated by other member node, this may ensures handling of resources properly and user satisfaction.

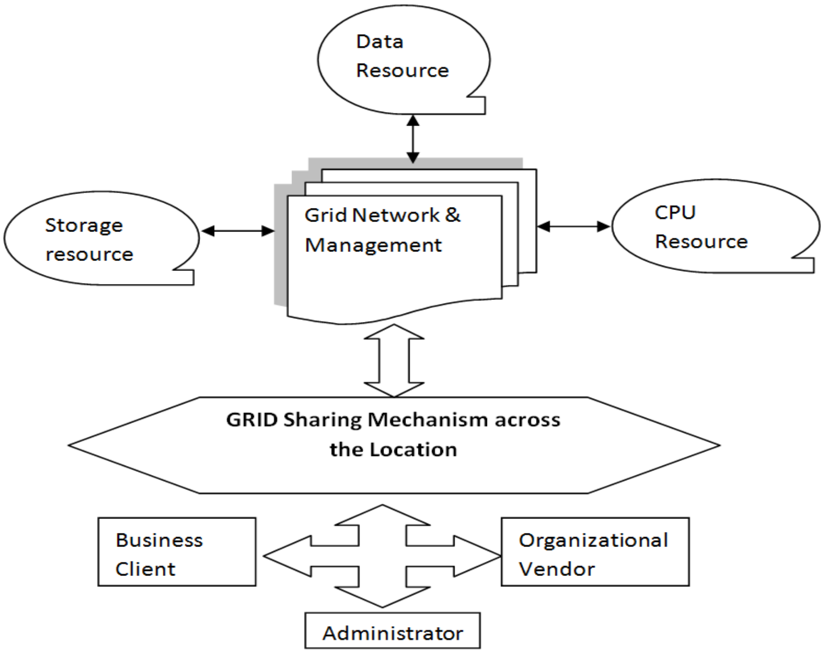


Figure 1.2.Grid Working Model

1.3. Cloud Computing

Cloud computing also a computational paradigm developed from distributed computing model lies between grid computing and supercomputing. The main benefit of cloud computing is services, it execute any type of job as a host service. These services are broadly classified into three categories: Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS). Here, all the computational resources are accessed as a utility and it executes the task very granularly when comparing with grid. Most of the organizations adopting cloud because of the benefits like self provisioning, elasticity, allocation needed resource which prevents unwanted installation of resources or idle resources. The working model of clod is shown in fig 1.3.

IaaS: Provides infrastructure facility to supply Virtual server and storage instances.

PaaS: Provides needed tools to develop their host on the infrastructure.

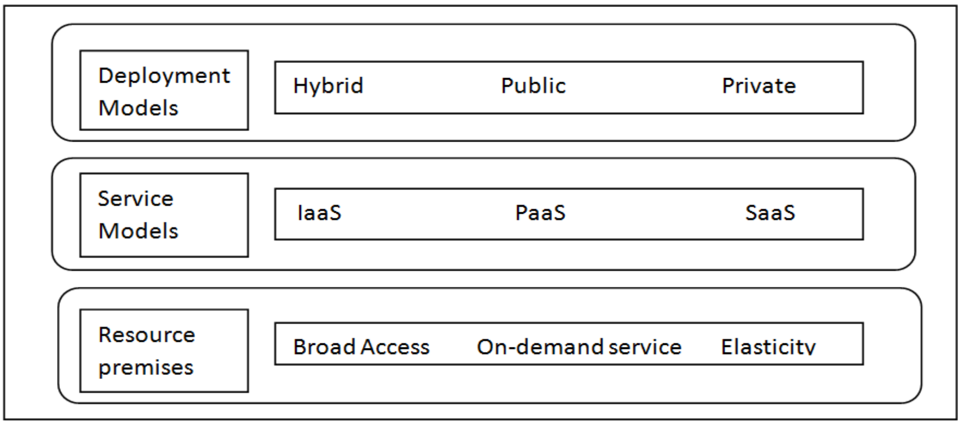
SaaS: It provides software applications which needed for the infrastructure.

Cloud computing allows easy and safety ways to migrate from any infrastructure model, this help organizations to adopt cloud computing. Cloud delivers the service into three models namely public cloud, private cloud and hybrid cloud. Based on the organizational security structure we can adopt the delivery model.

Public cloud: This model was used by third-party providers to utilize cloud services

Private cloud: this model was adopt by organizations to provide cloud service to the internal users

Hybrid cloud: This model has fusion property of public and private cloud models deployed to provide service for entire organizations. Sensitive and critical applications based organization adopts this delivery model.



Deployment Models	Hybrid	Public	Private
Service Models	IaaS	PaaS	SaaS
Resource premises	Broad Access	On-demand service	Elasticity

Figure 1.3.Cloud Deployment Model

1.4. Green Cloud Computing

This model is fully based on cloud environment with the adjustment on renewable resource usage, efficient usage of available resources with reduced carbon proficiency from both users as well as cloud providers. The root to attain green cloud computing is by applying full cloud premises on all working environment including data centers. Once we started to implement the procedures to reduce carbon emission from machines we reach the goal of green cloud. The model to achieve green environment is shown in fig 1.4.

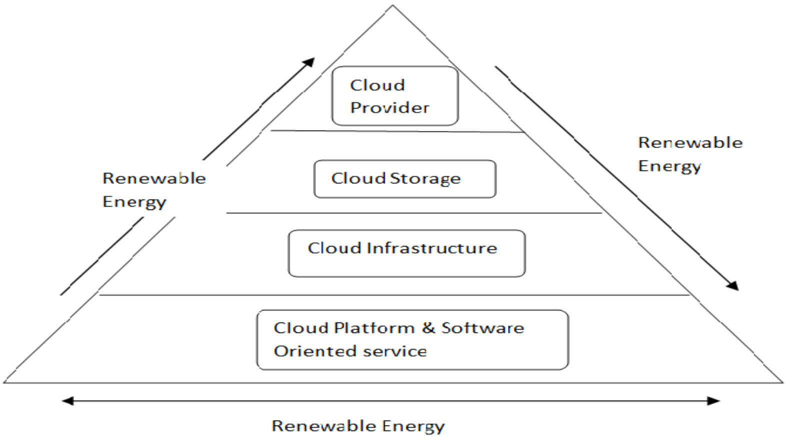


Figure 1.4. Green Cloud Computing

2. Load Balancer

Above mentioned distributed models are emphasize virtualized network access to the resources, they differ only on how they accessing the resource. The aim of these distributed models is no matter where is your resource, model direct you towards the resource available. This was done by the load balancer, which is used to distribute the request towards the resource and also helps to perform load distribution evenly towards the servers.

Load balancing can be referred as an optimization technique; helps to increase the throughput by utilizing available resources and reduce latency. It reduces redundancy by making unreliable system into reliable system by redirecting other server overloaded jobs by this it coupled with failover mechanism also. Without load balancing entire computing models suffer on managing the resource and in maintaining high availability of applications. The behavior of load balancer should be very generic, stable and possible to increase overhead to the system instantaneously. The basic working model of load balancer in represented in fig.2. Here we presented life history of load balancer and shown various mechanisms followed in advanced load balancer.

Basically, load balancing helps to ensure the following [3]:

Resources are easily available on demand.

Resources are efficiently utilized under the condition of high/low load.

Energy is saved in case of low load (i.e. when usage of cloud resources is below the certain threshold).

Cost of using resources is reduced.

Increase Responsiveness.

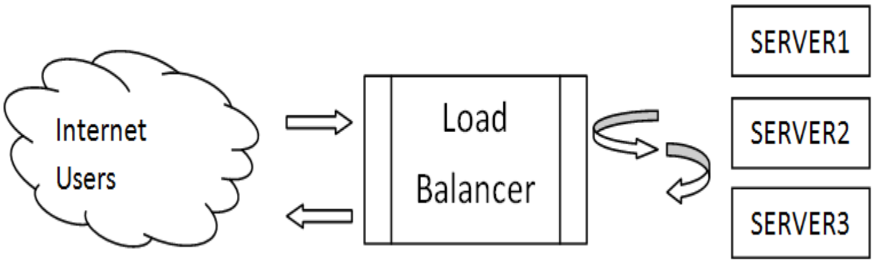


Figure 2. Load Balancer

2.1. Birth of Load Balancer

The Altering the idea of Domain Name System (DNS) record was the actual began of Load balancing functionality. Before load balancing to achieve the goal of scalability and high availability of resource, the commercial users adopt DNS only.

2.2. Domain Name System [DNS]

Wherever DNS is used to translate the human-readable domain names into number system (IP address) helps to recognize by machines. DNS scheme was altered and started to map more than one IP address to a particular domain host. By having multiple IP address on behalf of single domain name a multiple number of request can be sent to the server and get back the response from the servers. This offered simple load balancing structure held between small physical servers, but it was not working well for more number of requests. If number of request increases then the server fails on responding the request.

This method was not deterministic too, sudden failure of server may cause entire process to let down.

This was the real scenario for introducing a new technology “Load Balancing”.

The working mechanism of DNS was shown in fig 2.2.

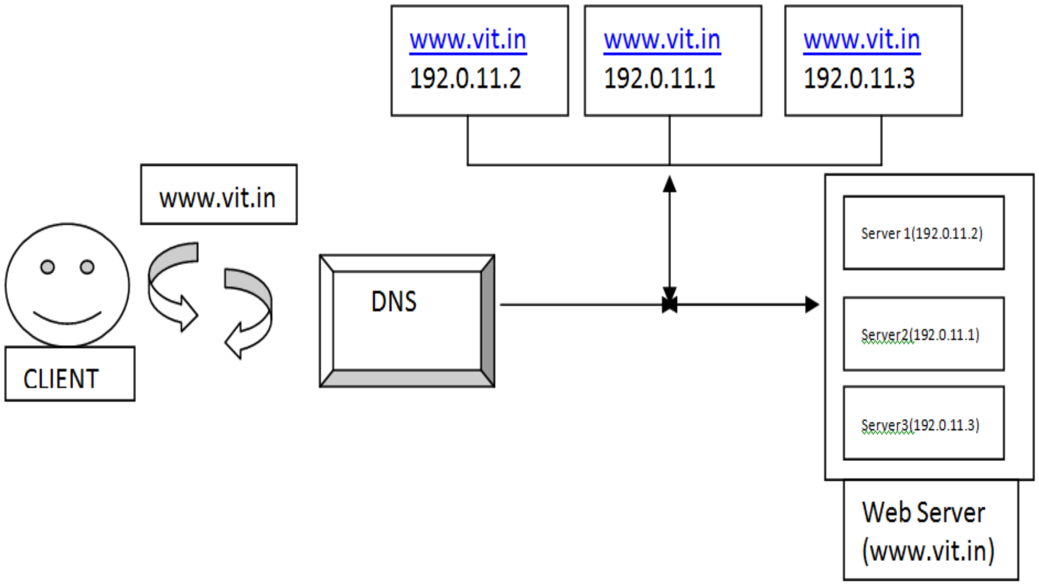


Figure 2.2. DNS Based Load Balancing

2.3. Formation of Load Balancers

Type Typically in next step the format of load balancers was changed by creating cluster software on applications based. Here the client was directed to single address which linked with specific cluster and the cluster contains several physical servers. So the cluster program decides which server serves best for the given application and route traffic over to the server. But this mechanism also helps only a little extent, later it faces issue as

It tends to work like an application specific and suits only for certain application.

This mechanism limited with the capability of physical server installed and number of clients processes the request.

2.4. Layer Based Load Balancer

Place In this, load balancing was performed with the help of protocols and layer functionalities, so layer 3/4 and layer 7 was chosen. In this stage, the load balancer is acts like application delivery controller were as layer 3/4 involves the distribution of traffic based on IP addresses, subnet. Layer 7 provides content based load balancing by considering URL, HTTP header and then distribute the traffic. After achieving layer based load balancing it provides way for new birth of several load balancer types.

Initially load balancers are used only for accepting n number of clients only, often it was turned to Application Delivery Controllers (ADC). Later on, years applying intelligent techniques on load balancers added more number of performance benefits to the internet users. The different types of load balancer exist in the history are follows.

Network based Load Balancer

Global Server Based Load Balancer.

2.4.1. Network Based Load Balancing

Place We can say this as second iteration technique on the history of load balancer; virtual servers are introduced between the client and server. Instead of connecting directly to the servers we can direct our request to the virtual machines, it connects to multiple sites which can respond the request on this basis the load was handled. It uses bi-directional network address translation (NAT) for forwarding the connection from client to outside world servers. NIC is used to establish communication between the outside world servers. The pictorial view of NBLB is shown below in fig.2.4.1. The drawbacks present in this mode as follows,

It lags on communicating with other clusters of outside world servers.

The overall performance of this type load balancer is not upto the level because it uses single NIC for entire system communication.

Mutual configuration is required

Do not support multiple schedule flows.

From the Network based load balancer, it ensures the functionalities like:

Central Authentication

Web Application Firewall (WAF)

Multi-Tenancy Support

Virtualization Options

Health Monitoring

Networking Integration

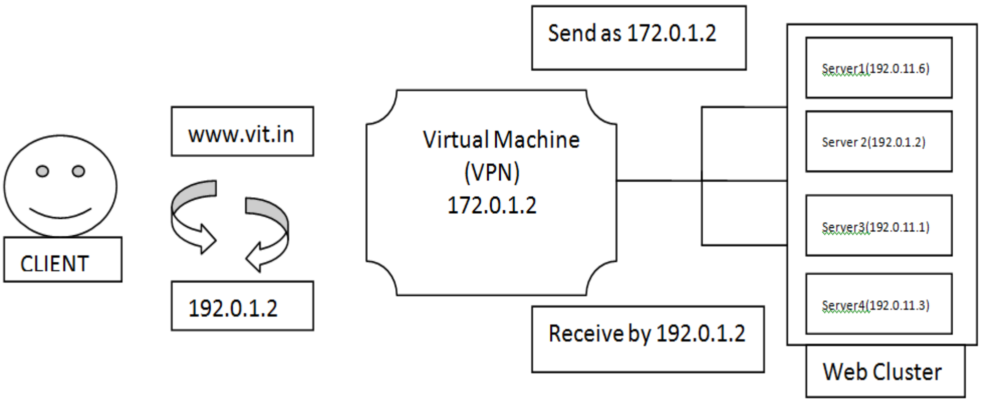


Figure 2.4.1. Network Based Load Balancer

2.4.2. Global Server Load Balancer

This was the basic load balancing concept used in the real time scenario. As mentioned in chapter 2 and figure 2 represent same as server load balancer. From here only a hierarchy was arise and several algorithms are proposed to increase the functionalities of load balancer. The term global server load balancer is derived because it distributes the load across various geographical locations instead of concentrating one web clusters and shown in fig.2.4.2. Naturally this type of load balancer works on wide area network (WAN) and basic server load balancer follows Local area network (LAN). From here only load balancer turned as server farm and started to provide service between two data centers.

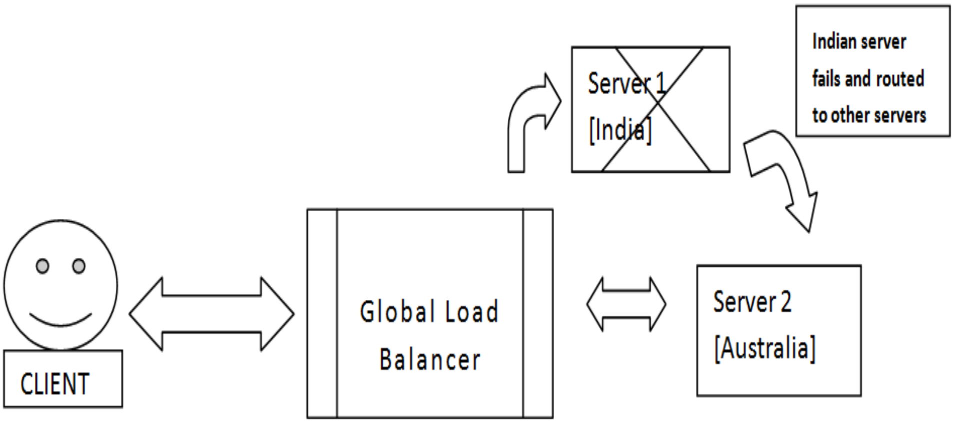


Figure 2.4.2. Global Load Balancer

The benefits of GSLB are stated below,

Because of cross country access and it maintains minimum latency by bringing content on the server very nearer to the user.

It provides high end redundancy in case of site fails. It diverts the traffic to all other remaining sites to bring the content to the user.

It supports maximum number of protocols used to carry internet actions.

It has effects to bring several machines to be appearing as one.

GLSB serves an important role in load distribution, even though some drawbacks are there GLSB practiced by many large scale system. The issues faced by GSLB are given below,

Persistence level is very low and the chances of having fail over systems are sometimes high

We can’t predict where the request is going to direct or which server the request is going to hit.

3.Load Balancing Algorithms

Depends upon the need of application the practices of load balancer also changes. So the metric used to distributing the traffic among the servers are also varied with user applications, this makes the users to apply different mathematical programs on SLB devices. Thus several algorithms are created, in this part we reviewed those algorithms and presented with their pros and consequence on it.

3.1. Algorithm Classification

On the basis of SLA policy, the load balancing algorithms are classified into static, dynamic and symmetric load balancing algorithms and the hierarchy was shown below in figure 3.1.

Static: It won’t consider the current state of the system or were the system standing. It precedes its algorithm only by considering the prior knowledge of the system.

Dynamic: It applies the mechanism by considering current system information. Prior knowledge of system state is not needed.

Symmetric: It works on the basis of sender and receiver initialization. It considers the priority of sender, receiver and transfer the load. When they need they can initialize any type of algorithm.

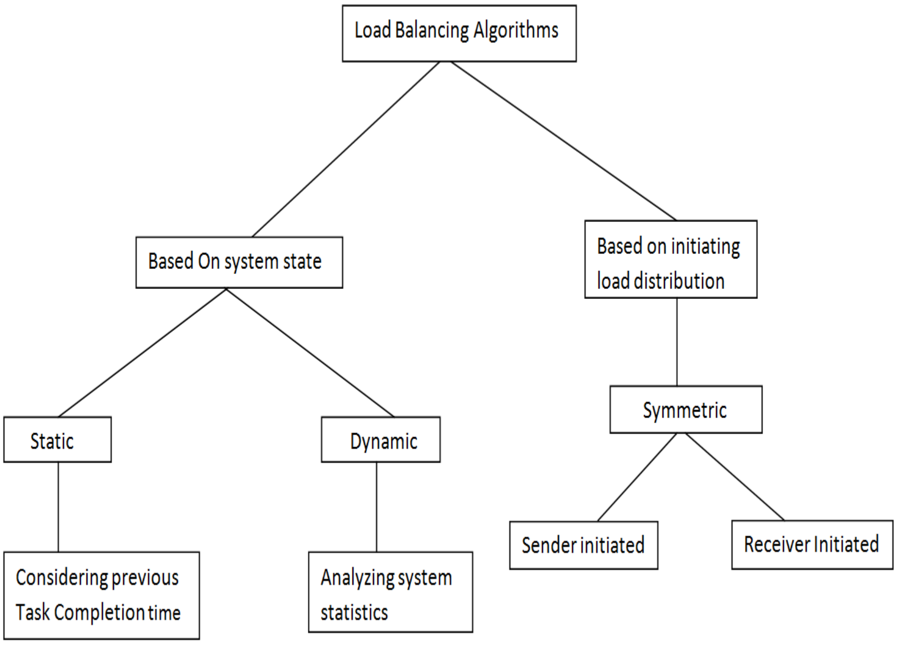

Load Balancing Algorithms
Based On system state
Based on initiating load distribution
Static
Dynamic
Symmetric
Considering previous Task Completion time
Analyzing system statistics
Sender initiated
Receiver Initiated

Figure 3.1.Algorithm Categorization

From the approach, we conclude Dynamic load balancing algorithms serve much better than static load balancing algorithms and the proposed algorithms on these approaches are discussed below in next part.

3.2. Works carried on Load Balancing Algorithms

In this section, we reviewed most practicing load balancing algorithms on distributed systems. Depends upon the classification we initially shown static load balancing algorithms and in second part we shown dynamic load balancing algorithms.

A. Static load Balancing Algorithms

This type of algorithms is suitable for small distributed environment with supporting processing capacity. The process inside static load balancing is carried with negligible communication delay and high in execution time. For these applications only static based algorithms are fit and best practicing algorithms are shown below.

4.1. Round Robin Algorithm [4]:

This was the former algorithm applied on load balancer, works on the basis of simply assigning the load with time interval and without considering the server parameters. Either the assigned load is completed or not, the load balancer continuing in assigning new task to the servers. The entire assigned task will be executed only on the basis of time interval only, once specified time interval was over the next job will be pushed into the progress for a particular time interval, whether the task is completed or not new task will be assigned to the machine.

The working flow of Round Robin algorithm is very simple, once the load balancer received the task it initiate the process by selecting the node (physical servers) and assigns the task events to the selected node with interval time. Depends upon the task node selection can be extended. Maximum this algorithm utilizes the nodes present in the system and final results are collected by communicating with the nodes. Under round robin method, even the node is overloaded it without considering it started to assign the new task, this decreases the performance of the system. And if a single node is overloaded and forms queue in executing the jobs will affect the entire system environment.

Interprocess communications among the nodes bring advantage to the system.

No need to maintain additional logs on server status.

Nodes are highly overloaded which degrade the performance. Distribution of load is not even.

4.1.1. Weighted Round Robin Algorithm [4]:

This is modified version of round robin algorithm in which limits are fixed on assigning load to the server. In round robin algorithm, nodes are overloaded that collapse the entire system performance. Here that problem was overcome by assigning weightage to the nodes which prevent overloaded. Upto to a certain extent only the system able to assign the task once it’s over another set of new task to be assigned.

Due to time interval, small unit jobs have to wait upto completion of other lengthy jobs.

Intercommunication is very robust.

4.2. Opportunistic Load Balancing Algorithm [5]:

This algorithm also most practiced algorithm in the history of load balancer and stand as competitive algorithm for round robin. The behavior of opportunistic algorithms is quite opposite to round robin algorithm were task are assigned to the node randomly. This algorithm mainly deals with lengthy task only and helps to execute the unexecuted task on the Load balancer. It schedules the unexecuted task without considering the present workload and execution time of the node.

Due to random load assignment, time taken to execute the load is very high.

Performance level is not appreciable.

4.3. Min-Min Load balancing Algorithm [6]

This algorithm considers the server constraint like execution time and job count before assigning the job. The scheduling of min-min algorithm gives priority to the jobs having minimum execution time. The time taken to execute the jobs waiting in the queue are calculated and maintained in node manager which was supervised by load balancer. This algorithm works well for the task with minimum execution time rather than the task with maximum execution time. So the task with maximum execution time has to wait for a time upto execution of all short unit jobs. In order to maintain this work flow, the server status should be updated periodically.

It ends with starvation of maximum time jobs which take more time to execute at the end of the moment and leads to decrease the performance.

Not suit for all type of system.

4.4. Max-Min Load balancing Algorithm [7]

The working of max-min algorithm appears same as min-min algorithm, instead of assigning minimum executing jobs giving priority to task having maximum execution time. Once the completion of lengthy jobs, task with minimum execution time will be assigned to the processor or else the short unit jobs have to wait upto the completion of all lengthy tasks. This algorithm also maintains time sequences about the task engaged and update the execution time periodically to the load balancer as well as processor.

Even very small unit of jobs has to wait for a long time upto completion of task with maximum execution time.

Performance level is very worst.

B. Dynamic Load Balancing Algorithms

Dynamic load balancing algorithms are widely used for large distributed environments and their results are more accurate comparing with static load balancing algorithms. By having cumulative information about the node is required to implement the dynamic load balancing algorithms. Depends upon the information only it assigns and reassigns the task to the nodes. The dynamic based algorithms are explained below.

4.5. Ant Colony Optimization Based Load Balancing Algorithm [8]

This algorithm provides optimized ways to schedule the load among the nodes. The working model of Ant colony algorithm is based on the action of ant towards seeking the food. It helps to distribute the load among the nodes effectively. How ant finding the path to carry the food to its colony, on the same basis the tasks are assigned to the nodes which serves better processing when comparing with other nodes present in it. Because of dynamic algorithm, it collects knowledge about the prior activities of a node. It helps the load balancer to find optimized node for assigning the jobs. Once the job assigned the second stage initiated to collect result from the individual node and combined to form a complete report. So assigning the jobs and collecting the results to be continued upto the completion of jobs.

This algorithm works efficiently.

The process involved is very tedious and not suit for large environments.

4.6. Honeybee Foraging Load balancing Algorithm [9]

This algorithm based on the working model of Honey bees. On that basis, this algorithm serves two functions (i).Searching the node,(ii)repeating the process. This algorithm also provides better optimization for distributing the load among the nodes. Once the job reaches the load balancer it started to search the better node to process the task. By having the job queues it search the node individually and find out the best node fit to the present job queues. This algorithm repeats the same process for all the jobs and then assign to the nodes. The difference between other algorithm and honey bee is it search the node on the basis of task, remaining algorithms search the node which has minimum load alone never consider about nature of the job.

Server waiting time has been reduced.

The results are not optimized due to lengthy procedure.

4.7. Biased Random Sampling load balancing algorithm [10]

This algorithm uses Virtual graph representation to represent the nodes and resources as vertex and degree. The entire setup of load balancer their connected nodes are considered as graph points, from the graph the distance of vertex or nodes connected with the load balancer is identified depends on the distance optimal path has been decided for each vertex, this also called as walk length. Once optimal path was decided it started to allocate the job to the nodes then the degree value is incremented depends on load allocation. If execution of job was over then degree value will be decremented. Additional feature of this algorithm is it set the threshold value for each node in order to avoid overload condition which helps to increase the system performance. Based on the degree value and walk length the load balancer compare and assign the job, if two parameters are not supported on assigning the task, then this algorithm randomly choose one neighbor node and started to assign the job. By this functionality, it is termed as biased random sampling.

By maintaining threshold value job execution by server is very fast when comparing with other techniques.

Walk length and random selection of node may bring disappointments.

8. Active Clustering Load Balancing Algorithm [11]

This method is upgraded version of biased random balancing algorithm. This type of algorithms are also called as match maker algorithm, introduces the concept of clustering. In biased random randomly chooses the node and assign job but here nodes are grouped with similar functionality node and execute the job once the job was over depends on next job type it started to assign new group disconnected from initial node and join with new node. This process is repeated again and again until jobs gets over.

Clustering of nodes increases utilization of resources this results increase in throughput.

Grouping the nodes based on the job type is difficult.

4.9. Genetic based Load Balancing Algorithm [12]

This algorithm is typically different when comparing with other algorithms, very first algorithm deal with global optimization is genetic model load balancing algorithm only. This algorithm uses stochastic searching technique to identify the best fit nodes for particular job, but searching bring difference to other algorithms here searching is very vast whereas other algorithm go for a local optimum solution only. This algorithm involves three steps inorder to meet load balancing namely selection, operation and replacement. By applying stochastic searching technique it find maximum node pairs fit for particular job, second it start to execute the job by applying necessary operations because of individual node pairs the problem also divided to each node pair and result are correlated finally from each nodes. This makes job execution very fast and increases performance of the system. Once the job was over, replacement of node pair takes place to form new pairs for new jobs by applying global searching.

Global optimization by stochastic searching added new strength to load balancer.

Throughput is very high.

Finding multiple pairs of nodes for single job executions leads to slight degradation in performance of the system.

Rescheduling of job is highly critical to apply.

4.10. Agent based load balancing algorithm [13]

This algorithm involves virtual machines as in the place of nodes and serves several agents like load agent, channel agent, migration agent. This algorithm introduces automated service on load balancing technique by initiating agents to complete the working activities involved in it. Once the load balancer received the job request it started the process of finding the virtual machine which doesn’t contain any job or having minimum load and able to perform current job is done by load agent. Next finding the optimal path to reach the virtual machine is done by channel agent. Later analyzing the next VM for new job and rescheduling the assigned jobs are carried by Migration agent.

Because of automated service throughput is very high.

Internal communication is very needed for this algorithm. This brings trouble on large systems.

Processing the agent itself need separate governance technique.

5. Result Analysis

From the review we developed a table which comprises the overall functionalities of those algorithms as shown below.

Table 7. Comparison of Load Balancing Algorithms

	Round Robin	Weigh . round robin	OLB	Min-min	Max-min	Ant colony	Honey bee	Biased random sampling	Active clustering	Genetic based	Agent based
Level of Implementation	Easy	Easy	Easy	medium	Medium	Hard	Hard	Hard	Hard	Hard	Hard
Req. of additional components	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Single point of failure	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
Scalability	No	No	NO	No	No	Yes	Yes	No	Yes	Yes	Yes
Throughput	Yes	Yes	No	No	No	Yes	Yes	No	No	No	No
Overall performance	Yes	Yes	No	No	No	Yes	Yes	Yes	No	Yes	Yes

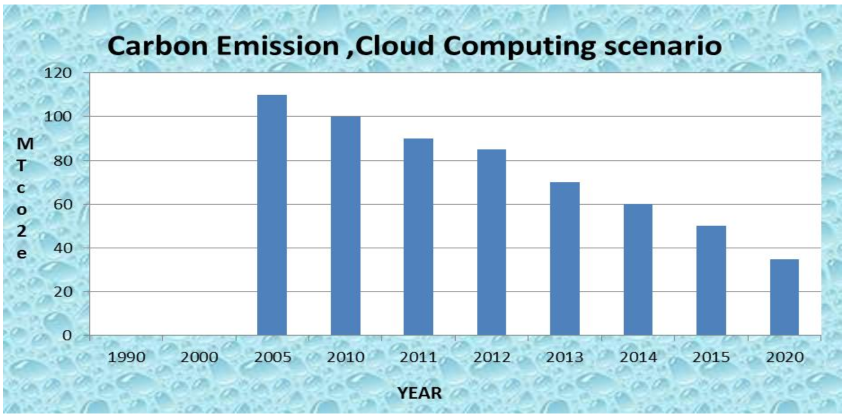
6. Achieving Green Computing

From the review, we can justify all the cloud load balancers works on following features,

Dynamic Provisioning: It supports on-demand services, prevents idle of resources and unwanted installation of new resource components.

Multi-tenancy: It uses the concept of central server and sharing. By this several cost cutting factors are enabled in computing infrastructure.

Increased Utilization: It helps to balance load requirement with less hardware and utilizing more server computational power. So green computing can be achieved from cloud computing strategy and the reduction of carbon emission is plotted below.



--MTCO2e-Metric Ton Carbon dioxide

Figure 6. Carbon Emission Graph

By cloud computing in 2020 the ICT regions set the goal as follows,

Table 6. Goal of ICT on Green Cloud

The cloud computing is Green Computing in 2020
o 90% reduction of Global Emission o 38% Reduction I Energy expenditures o 200 Million oil barrels to be save o 12.3 billion dollar of annual cost to be save

The ICT records are encouraging cloud computing inorder to reduce carbon emission control from computational servers. Before cloud computing, there don't have space to think about reducing carbon emissions, after the achievement of the above mentioned features brings the ideas into reality. By cloud computing not only computational effectiveness increased, economics of computational service also reached.

From the discussion, it clearly shows that n numbers of algorithms are proposed to balance the load evenly on server side. But none of the algorithms are satisfied the entire functional perspectives, we can observe this in the comparison table. To overcome the load balancing problem, we can follow apply migration service like IAAS on all server nodes instead of applying algorithms on the load balancer. By this, all the node can be acted like load balancer and results in fast computational execution.

7. Conclusion

Load balancing stands in irreplaceable position on cloud computing scenario. Another aspect shows increase in day by day usage of cloud the amount of carbon emission also multiplied. By maintaining proper load balancing algorithm helps to increase cloud performance and reduces carbon proficiency from that we can maintain dirty free computing. This type of system also helps us to increase utilization of available resources. In this paper we reviewed several load balancing algorithms and compared them on their performance wise and shown the tendency of load balancer to maintain the green effect on computational platforms.

References

[1] E. Altman, T. Başar, T. Jiménez and N. Shimkin, "Routing into two parallel links: game-theoretic distributed algorithms", Journal of Parallel and Distributed Computing., vol. 61, no. 9, **(2001)**, pp. 1367-1381.

[2] H. Kameda, "Optical Load Balancing in Distributed Computer Systems", Springer Science & Business Media, London, **(2012)**.

[3] M. Katyal and A. Mishra, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", International Journal of Distributed and Cloud computing, vol. 4, no.1,**(2014)**, pp.75-80.

[4] P. Samal and P. Mishra, "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing", IJCSIT, vol. 4, no. 3, **(2013)**, pp. 416-419.

[5] C. Kashyap, Dharmesh and J. Viradiya, "A Survey Of Various Load Balancing Algorithms In Cloud Computing", International Journal of Scientific & Technology Research, vol. 3, no. 11, **(2014)**, pp. 121-125.

[6] M. Y. Wu, W. Shu and H. Zhang, "Segmented Min-Min: A Static Mapping Algorithm for Meta-Tasks on Heterogeneous Computing Systems", Heterogeneous Computing Workshop, **(2000)**, pp. 375.

[7] T. Stützle and H. H. Hoos, "MAX–MIN ant system", Future generation computer systems., vol. 16, no. 8, **(2000)**, pp. 889-914.

[8] R. Mishra and A. Jaiswal, "Ant colony optimization: A solution of load balancing in cloud", International Journal of Web & Semantic Technology, vol. 3, no. 2 **(2012)**, p. 33.

[9] M. Randles, D. Lamb and A. T. Bendiab, "Experiments with Honeybee Foraging Inspired Load Balancing", Second International Conference on Developments in eSystems Engineering (DESE), **(2009)**; Abu Dhabi.

[10] C. Rahmeh, O. Abu, P. Johnson and A. T. Bendiab, "A dynamic biased random sampling scheme for scalable and reliable grid networks", INFOCOMP Journal of Computer Science, vol. 7, no. 4, **(2008)**, pp. 1-10.

[11] R. P. Padhy, “Load balancing in cloud computing systems", PhD diss., National Institute of Technology, Rourkela, **(2011)**.

[12] A. Singh, D. Juneja and M. Malhotra, "Autonomous Agent Based Load Balancing Algorithm in Cloud Computing", Procedia Computer Science, **(2015)**.

[13] R. Shimonski, Journal of Windows Server Clustering & Load Balancing, McGraw-Hill, Inc., **(2003)**.

[14] R. Tong and X. Zhu, "A load balancing strategy based on the combination of static and dynamic", 2nd International Workshop on Database Technology and Applications (DBTA), **(2010)**.

[15] K. G. Shin and Y. C. Chang, "Load sharing in distributed real-time systems with state-change broadcasts", Computers, IEEE Transactions, vol. 38, no. 8 **(1989)**, pp. 1124-1142.

[16] A. N. Tantawi and D. Towsley, "Optimal static load balancing in distributed computer systems", Journal of the ACM (JACM), vol. 32, no. 2, **(1985)**, pp. 445-465.

[17] S. Zhou and D. Ferrari, “An Experimental Study of Load Balancing Performance”, Proceedings of the 7th IEEE International Conference on Distributed Computing Systems, **(1987)**; California, USA.

[18] C. Grosan and A. Abraham, “Hybrid evolutionary algorithms: methodologies, architectures, and reviews", In Hybrid evolutionary algorithms, Springer Berlin Heidelberg, **(2007)**.

[19] D. L. Eager, E. D. Lazowska and J. Zahorjan, "Adaptive load sharing in homogeneous distributed systems", Software Engineering, IEEE Transactions, vol. 5, no. 8, **(1986)**, pp. 662-675.

[20] R. Buyya, C. S. Yeo and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities", 10th IEEE International Conference of High Performance Computing and Communications, HPCC, Dalian, **(2008)**.

[21] R. Bianchini and R. Rajamony, "Power and energy management for server systems", Computers, vol. 37, no. 1 ,**(2004)**, pp. 68-74.

Authors

G.Siva Shanmugam (b 1987), working as Assistant Professor at School of Computer Engineering at VIT university, Vellore, TN, India. His research interest includes Distributed systems, Web Security, Sensor Networks, Mobile and Internet Computing. He has had 6 years of teaching experience and currently doing his PhD research on Green Cloud computing.

N. Ch. S. N. Iyengar (b 1961) currently senior Professor at School of Computer Engineering at VIT University, Vellore, TN, India. His research interests include Wireless networks, Security Models, Agent based Intelligent Computing, and Cloud Computing. He has had 31 years of teaching and 20 years of research experience with a good number of publications in reputed International Journals & Conferences He is an Editorial Board member for many International Journals like ***Int. J. of Advances in Science and Technology,*** of SERSC, *Cybernetics and Information Technologies* (CIT)-Bulgaria, Egyptian Computer Science Journal-Egypt, IJConvC of Inderscience-China, IJCA (USA) etc., Also Editor in Chief for International Journal of Software Engineering and Applications(IJSEA) of AIRCC, Advances in Computer Science (ASC) of PPH, Guest editor for “Cloud Computing and Services” IJCNS. . He delivered Keynote/Invited talks at many International Conferences and workshops.