ABC - LOAD BALANCING TECHNIQUE - IN CLOUD COMPUTING

Miss. Neeta S. Nipane
Department of Computer Science and Engg
ACE,Nagthana Rd, Wardha(MH),INDIA
neetanipane@gmail.com

Prof. Nutan M. Dhande
Department of Computer Science and Engg
ACE,Nagthana Rd, Wardha(MH),INDIA
nutandhande@gmail.com

Abstract— Cloud Computing is an emerging area in the field of information technology (IT). Load balancing is one of the main challenges in cloud computing. It is a technique which is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overloaded. Load balancing techniques help in optimal utilization of resources and hence in enhancing the performance of the system. The goal of load balancing is to minimize the resource consumption which will further reduce energy consumption and carbon emission rate that is the dire need of cloud computing. This determines the need of new metrics, energy consumption and carbon emission for energy-efficient load balancing in cloud computing. This paper mainly focused on the concept of load balancing technique in cloud computing, the existing load balancing techniques and also discusses the different qualitative metrics or parameters like performance, scalability, associated overhead etc.

Keywords— Load Balancing, Green Computing, Carbon Emission, Dynamic Load Balancing, Workload and Client aware policy (WCAP),etc.

I. INTRODUCTION

Cloud computing is an on demand service in which shared resources, information, software and other devices are provided according to the clients' requirement at specific time. It's a term which is generally used in case of Internet. The whole Internet can be viewed as a cloud. Capital and operational costs can be cut using cloud computing.

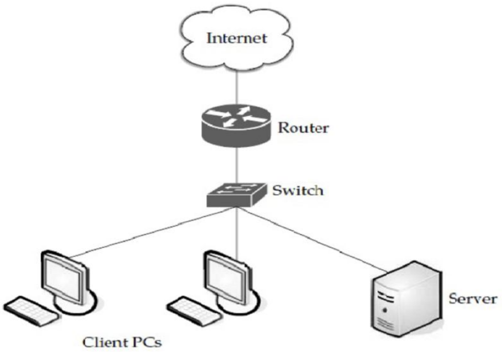

Figure 1: A cloud is used in network diagrams to depict the Internet [1].

Load balancing in cloud computing systems is really a challenge now. Jobs can't be assigned to appropriate servers and clients individually for efficient load balancing as cloud is a very complex structure and components are present throughout a wide spread area. Here some uncertainty is attached while jobs are assigned different performance parameters like throughput, latency etc. for the clouds of different sizes. As the whole Internet can be viewed as a cloud of many connection-less and connection oriented services, thus concept of load balancing in Wireless sensor networks (WSN) proposed in can also be applied to cloud computing systems as WSN is analogous to a cloud having no. of master computers (Servers) and no. of slave computers (Clients) joined in a complex structure. A comparative study of different algorithms has been carried out using divisible load scheduling theory proposed. In case of Cloud computing services can be used from diverse and widespread resources, rather than remote servers or local machines. There is no standard definition of Cloud computing. Generally it consists of a bunch of distributed servers known as masters, providing demanded services and resources to different clients known as clients in a network with scalability and reliability of datacenter.

1.1 Cloud Components

A Cloud system consists of 3 major components such as clients, datacenter, and distributed servers. Each element has a definite purpose and plays a specific role.

i. Clients : End users interact with the clients to manage information related to the cloud. Clients generally fall into three categories as given in [1]:

- Mobile: Windows Mobile Smartphone, smartphones, like a Blackberry, or an iPhone.
- Thin: They don't do any computation work. They only display the information. Servers do all the works for them. Thin clients don't have any internal memory.

- Thick: These use different browsers like IE or mozilla Firefox or Google Chrome to connect to the Internet cloud. Now-a-days thin clients are more popular as compared to other clients because of their low price, security, low consumption of power, less noise, easily replaceable and repairable etc.

ii) Datacenter : Datacenter is nothing but a collection of servers hosting different applications. A end user connects to the datacenter to subscribe different applications.

iii) Distributed Servers : Distributed servers are the parts of a cloud which are present throughout the Internet hosting different applications. But while using the application from the cloud, the user will feel that he is using this application from its own machine.

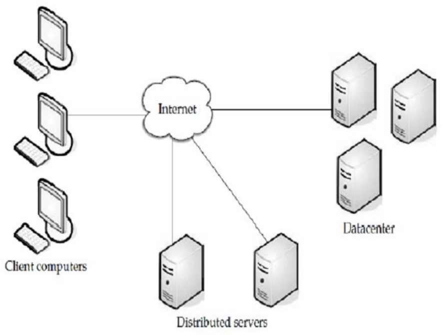

Figure 2: Three components make up a cloud computing solution [1].

1.2 Type of Clouds

Based on the domain or environment in which clouds are used, clouds can be divided into 3 catagories :

- Public Clouds
- Private Clouds
- Hybrid Clouds (combination of both private and public clouds)

1.3 Load Balancing

Load balancing is one of the major issues in cloud computing [4]. It is a mechanism which distributes the dynamic local workload evenly across all the nodes in the whole cloud. This will avoid the situation where some nodes are heavily loaded while others are idle or doing little work. It helps to achieve a high user satisfaction and resource utilization ratio. Hence, this will improve the overall performance and resource utility of the system. It also ensures that every computing resource is distributed efficiently and fairly [5]. It further prevents bottlenecks of the system which may occur due to load imbalance. When one or more components of any service fail, load balancing helps in continuation of the service by implementing fair-over. However, resource consumption can be kept to a minimum with proper load balancing which not only helps in reducing costs but making enterprises greener [6] [5]. Scalability which is one of the very important features of cloud computing is also enabled by load balancing. Hence, improving resource utility and the performance of a distributed system in such a way will reduce the energy consumption and carbon footprints to achieve Green computing [7] [8] [9]. Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine. Load balancing is done so that every virtual machine in the cloud system does the same amount of work throughout therefore increasing the throughput and minimizing the response time. Load balancing is one of the important factors to heighten the working performance of the cloud service provider. Balancing the load of virtual machines uniformly means that anyone of the available machine is not idle or partially loaded while others are heavily loaded. One of the crucial issue of cloud computing is to divide the workload dynamically.

1.4 Why Balancing in Cloud Computing

Load balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio, making sure that no single node is overwhelmed, hence improving the overall performance of the system. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. It also helps in implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning, reducing response time etc. Load balancing is also needed for achieving Green computing in clouds . The factors responsible for it are:

Limited Energy Consumption : Load balancing can reduce the amount of energy consumption by avoiding over hearting of nodes or virtual machines due to excessive workload .

Reducing Carbon Emission : Energy consumption and carbon emission are the two sides of the same coin. Load balancing helps in reducing energy consumption which will automatically reduce carbon emission and thus achieve Green Computing .

1.5 Goals of Load balancing

The goals of load balancing are:

- To improve the performance substantially

- To have a backup plan in case the system fails even partially
- To maintain the system stability
- To accommodate future modification in the system

1.7 Types of Load balancing algorithms

Depending on who initiated the process, load balancing algorithms can be of three categories as given in [3]:

- Sender Initiated: If the load balancing algorithm is initialized by the sender
- Receiver Initiated: If the load balancing algorithm is initiated by the receiver
- Symmetric: It is the combination of both sender initiated and receiver initiated

Depending on the current state of the system, load balancing algorithms can be divided into 2 categories as given in [3]:

Static Algorithm

Static algorithms divide the traffic equivalently between servers. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there are lots of problems appeared in this algorithm. Therefore, righted round robin was defined to improve the critical challenges associated with round robin.

Dynamic Algorithm

Dynamic algorithms designated proper rights on servers and by searching in whole network a lightest server preferred to balance the traffic. However, selecting an appropriate server needed real time Communication with the networks, which will lead to extra traffic added on system. No prior knowledge is needed. So it is better than static approach. Here discuss on various dynamic load balancing algorithms for the clouds of different sizes.

In a distributed system, dynamic load balancing can be done in two different ways: distributed and non-distributed. In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. A benefit, of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt; it instead would affect the system performance to some extent. Distributed dynamic load balancing can introduce immense stress on a system in which each node needs to interchange status information with every other node in the system. It is more advantageous when most of the nodes act individually with very few interactions with others.

In non-distributed type, either one node or a group of nodes do the task of load balancing. Non-distributed dynamic load balancing algorithms can take two forms: centralized and semi-distributed. In the first form, the load balancing algorithm is executed only by a single node in the whole system: the central node. This node is solely responsible for load balancing of the whole system. The other nodes interact only with the central node. In semi-distributed form, nodes of the system are partitioned into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in each cluster by appropriate election technique which takes care of load balancing within that cluster. Hence, the load balancing of the whole system is done via the central nodes of each cluster. Centralized dynamic load balancing takes fewer messages to reach a decision, as the number of overall interactions in the system decreases drastically as compared to the semi distributed case. However, centralized algorithms can cause a bottleneck in the system at the central node and also the load balancing process is rendered useless once the central node crashes. Therefore, this algorithm is most suited for networks with small size.

1.8 Policies or Strategies in dynamic load balancing

The different policies in dynamic load balancing are:

- **Transfer Policy:** The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote is referred to as Transfer policy or Transfer strategy.
- **Selection Policy**: It specifies the processors involved in the load exchange (processor matching)
- **Location Policy :** The part of the load balancing algorithm which selects a destination node for a transferred task is referred to as location policy or Location strategy.

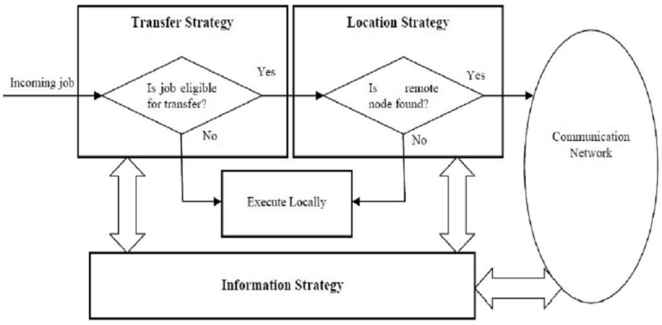

Figure 3: Interaction among components of a Dynamic Load Balancing Algorithm

- **Information Policy :** The part of the dynamic load balancing algorithm responsible for collecting information about the nodes in the system is referred to as Information policy or Information strategy.

- **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.
- **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.

II. LITERATURE REVIEW

2.1 Existing Load Balancing Techniques in Cloud Computing

COMPARISON OF LOAD BALANCING ALGORITHMS

Table 1 shows the comparison of LB algorithms which were discussed above.

Comparison of LB Algorithms Algorithm	**Description**	**Advantages**
Dynamic Round Robin Algorithm	1. Uses two rules to save the power consumption 2. Works for consolidation of VM	Reduce the power consumption
Decentralized Content Aware Load Balancing Algorithm	1. Uses Unique and Special Property(USP) of nodes 2. Uses content information to narrow down the search	1. Improves the searching performance hence increasing overall performance 2. Reduces idle time of nodes
Join-Idle Queue Algorithm	1. Assigns idle processors to dispatchers for the availability of idle processors 2. Then assigns jobs to processors to reduce average queue length	1. Reduces system load 2. Less communication overhead
Honeybee Foraging Behavior(ABC)	Achieves global load balancing through local server actions	Improved scalability

III. PROBLEM STATEMENT

The concept of introducing constrained handling procedure to the original ABC to tackle constrained optimization problems. ABC algorithm was also applied to solve large scale optimization problems. Karaboga and Akay modified the ABC algorithm to solve constrained optimization problems. The performance of ABC algorithm with the integration of Greedy Randomized Adaptive Search Heuristic and shift neighborhood structures for a generalized assignment problem was investigated. ABC was modified by the authors through the integration of the employed and onlooker phases with shift neighborhood structures applied sequentially.

IV. FUNDAMENTALS TO THE ARTIFICIAL BEE COLONY

A. Artificial Bee Colony: Analogy

The ABC algorithm is a swarm based, meta-heuristic algorithm based on the model first proposed by [11] on the foraging behaviour of honey bee colonies. The model is composed of three important elements: employed and unemployed foragers, and food sources. The employed and unemployed foragers are the first two elements, while the third element is the rich food sources close to their hive. These behaviors are necessary for self organization and collective intelligence: recruitment of forager bees to rich food sources, resulting into positive feedback and simultaneously, the abandonment of poor sources by foragers, The ABC consists of three groups of artificial bees: employed foragers, onlookers and scouts. The employed bees comprise the first half of the colony whereas the second half consists of the employed bees are linked to particular food sources. The onlookers observe the dance of the employed bees within the hive, to select a food source, whereas scouts search randomly The search cycle of ABC consists of three rules: (i) sending the employed bees to a food source and evaluating the nectar quality; for new food sources. causes negative feedback [10]. (ii) onlookers choosing the food sources after obtaining information from employed bees and calculating the nectar quality; (iii) determining the scout bees and sending them onto possible food sources. The positions of the food sources are randomly selected by the bees at the initialization stage and their nectar qualities are measured. The employed bees then share the nectar information of the sources with the bees waiting at the dance area within the hive. After sharing this information, every employed bee returns to the food source visited during the previous cycle, since the position of the food source had been memorized and then selects another food source using its visual information in the neighbourhood of the present one. At the last stage, an onlooker uses the information obtained from the employed bees at the dance area to select a food source. The probability for the food sources to be selected increases with increase in its nectar quality. Therefore, the employed bee with information of a food source with the highest nectar quality recruits the onlookers to that source. It subsequently chooses another food source in the neighborhood of the one currently in her memory based on visual information (i.e. comparison of food source positions).

A new food source is randomly generated by a scout bee to replace the one abandoned by the onlooker bees. This search process could be represented in algorithm (1) as follows:

Algorithm 1 Schematic pseudocode of ABC procedure

```
Initialize the ABC and problem parameters
Initialize the Food Source Memory (FSM)    repeat
    Send the employed bees to the food sources.
    Send the onlookers to select a food source.
    Send the scouts to search possible new food.
    Memorize the best food source.
until (termination criterion are met)
```

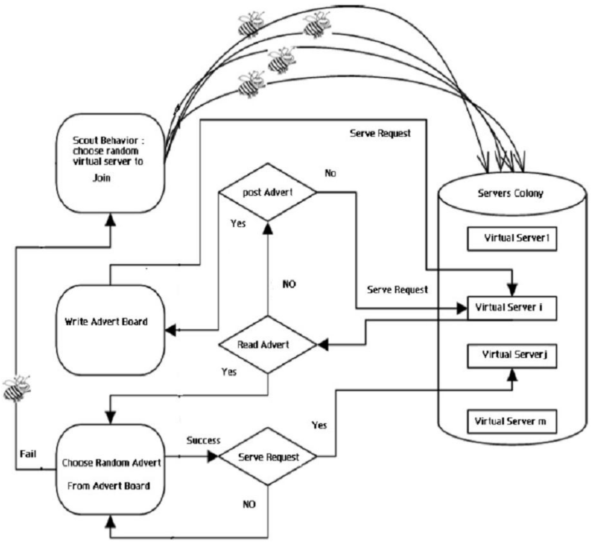

Figure 4: Server Allocations by Foraging in Honey bee technique

4.2 Qualitative Metrics for Load Balancing

In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio. The different qualitative metrics or parameters that are considered important for load balancing in cloud computing are discussed as follows:

i) Throughput: The total number of tasks that have completed execution is called throughput. A high throughput is required for better performance of the system.

ii) 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.

iii) 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.

iv) 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.

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

4.3 Load Balancing Challenges In The Cloud Computing

Although cloud computing has been widely adopted. Research in cloud computing is still in its early stages, and some scientific challenges remain unsolved by the scientific community, particularly load balancing challenges.

i) Automated Service Provisioning: A key feature of cloud computing is elasticity, resources can be allocated or released automatically. How then can we use or release the resources of the cloud, by keeping the same performance as traditional systems and using optimal resources?

ii) Virtual Machines Migration: With virtualization, an entire machine can be seen as a file or set of files, to unload a physical machine heavily loaded, it is possible to move a virtual machine between physical machines. The main objective is to distribute the load in a datacenter or set of datacenters. How then can we dynamically distribute the load when moving the virtual machine to avoid bottlenecks in Cloud computing systems?

iii) Energy Management: The benefits that advocate the adoption of the cloud is the economy of scale. Energy saving is a key point that allows a global economy where a set of global resources will be supported by reduced providers rather that each one has its own resources. How then can we use a part of data enter while keeping acceptable performance?

iv) Stored Data Management: In the last decade data stored across the network has an exponential increase even for companies by outsourcing their data storage or for individuals, the management of data storage or for individuals, the management of data storage becomes a major challenge for cloud computing. How can we distribute the data to the cloud for optimum storage of data while maintaining fast access.

V. APPLICATION OF ABC BY AREA OF DISCIPLINE

Many applications of ABC algorithm to real world and benchmark optimization problems have been reported, whereas substantial portion of the publications also compared the performance of ABC with other optimization algorithms. In the following subsections, some areas to which ABC was applied are discussed in detail. These areas include:

A. Benchmarking Optimization

Existing and new optimization techniques are evaluated using numerous benchmark problems that turned out to be de facto standards. Some examples of these optimization problems include continuous and discrete variables, constrained and unconstrained, and unimodal and multi-modal.

B. Bioinformatics application

In the field of computational biology and bioinformatics, ABC was utilized for protein structure prediction, using the three-dimensional hydrophobic polar model with side-chains (3DHP-SC).

D. Clustering and Mining Applications

Data clustering problems were previously tackled using a variety of information technology (IT) approaches. The fundamental focus of clustering is to split a data set into clusters, such that there is a high relationship between the elements within a cluster, but a low relationship between the elements of different clusters. In this regard, ABC was utilized for sensor deployment problem .

E. Image processing Applications

Several difficult problems exist in pattern recognition and image processing research areas. Searching for efficient optimization algorithm to address these problems has been the focus of much of active research.

VI. CONCLUSION AND FUTURE SCOPE

Based on the papers considered however, it could be observed that substantial part of the research was concentrated towards modifying and hybridizing ABC to solve diverse sets of problems, including those on data clustering, engineering design problems, medical image processing, scheduling problem etc. The majority of applications developed or proposed were aimed at solving constrained and unconstrained optimization problems.

The ABC has come to be recognized as a powerful and robust global optimization algorithm, capable of tackling unimodal and multimodal, non-differentiable, non-linear objective functions. In conclusion, ABC remains a promising and interesting algorithm, which would continue to be extensively used by researchers across diverse fields. Its potential advantage of being easily hybridized with different meta-heuristic algorithms and components makes it robustly viable for continued utilization for more exploration and enhancement possibilities in many more years to come.

REFERENCES

[1] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW HILL Edition 2010.

[2] Martin Randles, David Lamb, A. Taleb-Bendiab, Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[3] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[4] B. P. Rima, E. Choi, and I. Lumb, “A Taxonomy and Survey of Cloud Computing Systems”, Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pages 44-51.

[5] R. Mata-Toledo, and P. Gupta, “Green data center: how green can we perform”, Journal of Technology Research, Academic and Business Research Institute, Vol. 2, No. 1, May 2010, pages 1-8.

[6] S. Kabiraj, V. Topka, and R. C. Walke, “Going Green: A Holistic Approach to Transform Business”, International Journal of Managing Information Technology (IJMIT), Vol. 2, No. 3, August 2010, pages 22-31.

[7] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, “Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport”, Proceedings of the IEEE, Vol. 99, No. 1, January 2011, pages 149- 167.

[8] Z. Zhang, and X. Zhang, “A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation”, Proceedings of 2nd International Conference on Industrial Mechatronics and Automation ICIMA), Wuhan, China, May 2010, pages 240- 243.

[9] Nidhi Jain Kansal , Inderveer Chana, “Cloud Load Balancing Techniques: A Step Towards Green Computing”, IJCSI, Vol. 9, Issue 1, January 2012.

[10] 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.

[11] H.Mehta,P. Kanungo, and M. Chandwani, “Decentralized content aware load balancing algorithm for distributed computing environments”, Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), February 2011, pages 370-375.