Usage of Dynamic Load Balancing for Distributed System in Cloud Computing

Reeta Mishra
Assistant Professor
K.J. Institute of Engineering & Technology, Savli, Vadodara, Gujarat

Abstract— Cloud computing involves virtualization, distributed computing, networking, software and web services. A cloud consists of several elements such as clients, datacenter and distributed servers. It includes fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Two of the main obstacles in the usage of cloud computing are Cloud Security and Performance stability. Performance stability can be improved by Load balancing. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. This technique can be sender initiated, receiver initiated or symmetric type (combination of sender initiated and receiver initiated types).

Key words: Load Balancing, Cloud Computing, Cloud Security, Performance Stability, Distributed System

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.

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, data storage, software applications and other computing services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

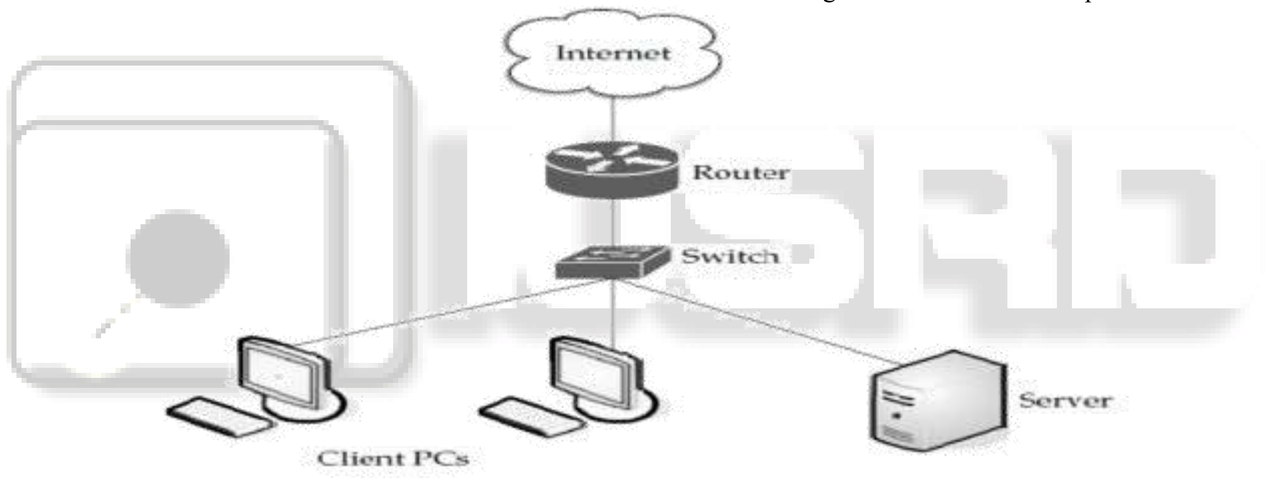

Fig. 1: A Cloud Is Used In Network Diagrams To Depict The Internet [1].

There are three services of cloud shown as below-

1) Infrastructure as a service (IaaS)
2) Software as a service (SaaS)
3) Platform as a service (PaaS)

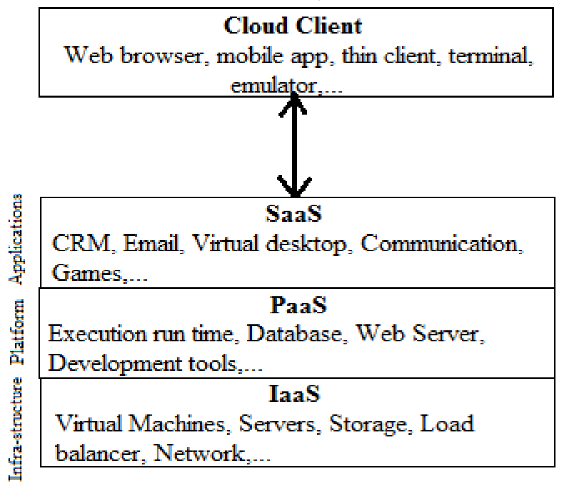
Cloud Client
Web browser, mobile app, thin client, terminal, emulator,...
Applications
SaaS
CRM, Email, Virtual desktop, Communication, Games,...
Platform
PaaS
Execution run time, Database, Web Server, Development tools,...
Infra-structure
IaaS
Virtual Machines, Servers, Storage, Load balancer, Network,...

Fig. 2: Cloud Services

II. Distributed File System

Distributed file system for cloud is a file system that allows many clients to have access to the same data/file providing important operations (create, delete, modify, read and write). Each file may be partitioned into several parts called chunks. Each chunk is stored in remote machines. Typically, data is stored in files in a hierarchical tree where the nodes represent the directories. Hence, it facilitates the parallel execution of applications. There are several ways to share files in a distributed architecture.

Meanwhile, the security and performance of the system must be ensured. Confidentiality, availability and integrity are the main keys for a secure system. Nowadays, users can share resources from any computer/device, anywhere and everywhere through internet thanks to cloud computing which is typically characterized by the scalable and elastic resources such as physical servers, applications and any services that are virtualized and allocated dynamically. Thus, synchronization is required to make sure that all devices are update. Distributed file systems enable also many big, medium and small enterprises to store and access their remote data exactly as they do locally, facilitating the use of variable resources.

III. Load Balancing

Load balancing in cloud computing systems is really a challenge now. It means distributing the amount of work to do between different servers in order to get more work done in the same amount of time and serve clients faster. Always a distributed solution is required. Because it is not always practically feasible or cost efficient to maintain one or more idle services just as to fulfill the required demands. 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.

IV. Challenges in Cloud Computing Load Balancing

Finding a solution for problems in load balancing is never an easy process there will be many challenges to be faced while developing a solution. Here in this section we will discuss some of the common challenges that might be faced while developing a solution for a problem of load balancing in cloud computing.

A. 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 [2]. So while developing a load balancing algorithm one should try for better results in spatially distributed nodes balancing the load effectively reducing network delays.

B. Migration Time:

In cloud computing the service-on-demand method will be followed which means when there is a demand for a resource the service will be provided to the required client. So while serving the client on his demands sometimes we need to migrate resources from long distances due to unavailability in near locations. In such cases the time of migration of the resources from far locations will be more which will affect the performance of the system.While developing an algorithm one should note that resource migration time is an important factor that greatly affects the performance of the system.

C. Performance of the System:

It doesn’t mean that if the complexity of an algorithm is high then the performance of the system will be high. Any time load balancing algorithm must be simple to implement and easy to operate. If the complexity of algorithm is high then the implementation cost will also be more and even after implementing the system performance will be decreased due to more delays in the functionality of the algorithm.

D. Failure of Controller:

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 algorithms when the controller fails the whole system 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 and distributed fashion so that when a node acting as a controller fails the system will not halt [5]. In such cases the control will be given to other nodes and they will act as controllers of the system.

E. 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. Even though by using energy efficient hardware architectures which slows down the processor speed and turn off machines that are not under use the energy management is becoming difficult. So to achieve better results in energy management a load balancing algorithm should be designed by following Energy Aware Job Scheduling methodology [6].

F. Security:

Security is one of the problems that cloud computing has in its top most priority. 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 must be strong enough to reduce the security attacks but should not be vulnerable.

V. Dynamic Load balancing algorithm

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. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative [4]. In the first one, the nodes work side-by-side to achieve a common objective, for example, to improve the overall response time, etc. In the second form, each node works in dependently toward a goal local to it, for example, to improve the response time of a local task. Dynamic load balancing algorithms of distributed nature, usually generate more messages. 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 effect 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.

A. *Policies or Strategies in Dynamic Load Balancing:*

There are 4 policies [4]:

1) Transfer Policy: The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote node is referred to as Transfer policy or Transfer strategy.
2) Selection Policy: It specifies the processors involved in the load exchange (processor Matching)
3) 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.
4) 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.

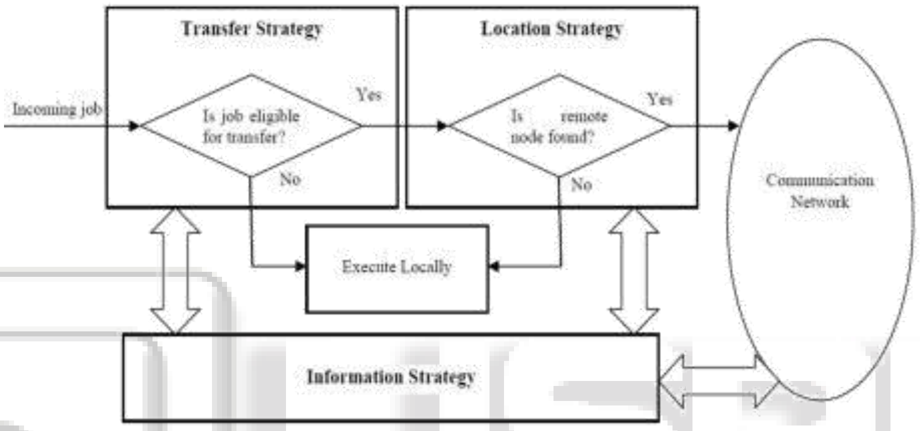

Fig. 3: Interaction among Components of a Dynamic Load Balancing Algorithm

VI. HONEYBEE FORAGING ALGORITHM

In case of load balancing, as the webservers 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. One measure of this reward can be the amount of time that the CPU spends on the processing of a request. The dance floor in case of honey bees is analogous to an advert board here. This board is also used to advertise the profit of the entire colony.

Each of the servers takes the role of either a forager or a scout. The server after processing a request can post their profit on the advert boards with a probability of pr. A server can choose a queue of a VS by a probability of px showing forage/explore behavior, or it can check for advertisements (see dance) and serve it, thus showing scout behavior. A server serving a request, calculates its profit and compare it with the colony profit and then sets its px. If this profit was high, then the server stays at the current virtual server; posting an advertisement for it by probability pr. If it was low, then the server returns to the forage or scout behavior.

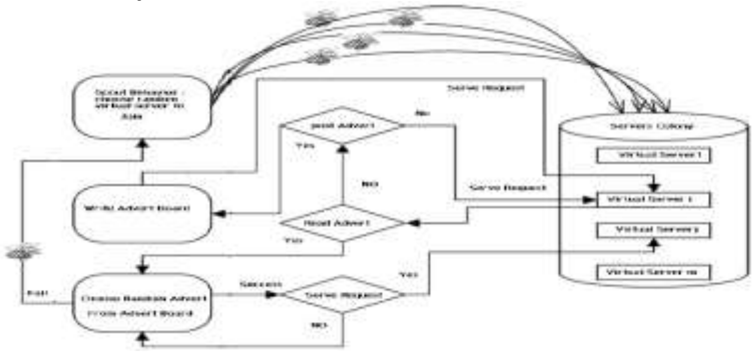

Fig. 4: Server Allocations by Foraging In Honey Bee Technique

A. *Biased Random Sampling:*

Here a virtual graph is constructed, with the connectivity of each node (a server is treated as a node) representing the load on the server. Each server is symbolized as a node in the graph, with each indegree directed to the free resources of the server. Regarding job execution and completion,

1) Whenever a node does or executes a job, it deletes an incoming edge, which indicates reduction in the availability of free resource.
2) After completion of a job, the node creates an incoming edge, which indicates an increase in the availability of free resource.

B. *Active Clustering:*

Active Clustering works on the principle of grouping similar nodes together and working on these groups. The process involved is:

1) A node initiates the process and selects another node called the matchmaker node from its neighbors satisfying the criteria that it should be of a different type than the former one.
2) The so called matchmaker node then forms a connection between a neighbor of it which is of the same type as the initial node.
3) The matchmaker node then detaches the connection between itself and the initial node.

VII. PROPOSED WORK

The distributed network may follow different topologies. The tasks are distributed over the whole network. One topological network connects with the other through a gateway. One of the physical topologies forming a cloud.

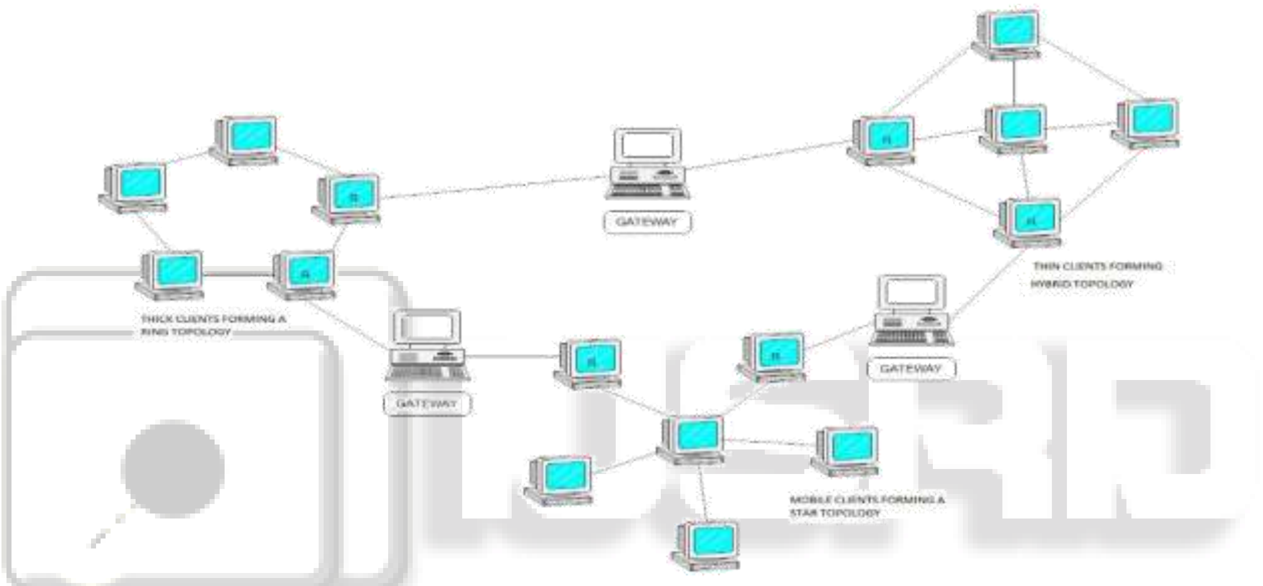
GATEWAY
THIN CLIENTS FORMING HYBRID TOPOLOGY
THICK CLIENTS FORMING A RING TOPOLOGY
GATEWAY
GATEWAY
MOBILE CLIENTS FORMING A STAR TOPOLOGY

Fig. 5:

In case of clouds is an optimal division of loads among a number of master computers, slave computers and their communication links. Our objective is to obtain a minimal partition of the processing load of a cloud connected via different communication links such that the entire load can be distributed and processed in the shortest possible amount of time.

The concept of load balancing in Wireless sensor networks (WSN) proposed in [9] can also be applied to clouds as WSN is analogous to a cloud having no. of Master computers (Servers) and no. of slave computers (Clients).The slave computers are assumed to have a certain measurement capacity. We assume that computation will be done by the master computers, once all the measured data is gathered from corresponding slave computers. Only the measurement and communication times of the slave computers are considered and the computation time of the slave computers is neglected. Here we consider both heterogeneous and homogeneous clouds. That is the cloud elements may possess different measurement capacities, and communication link speeds or the same measurement capacities, and communication link speeds. One slave computer may be connected to one or more master computers at a certain instant of time. In case of clouds, an arbitrarily divisible load without having any previous Relations is divided and first distributed among the various master computers (for simplicity here the load is divided equally between the master computers) and the each master computer distributes the load among the corresponding slave computers so that the entire load can be processed in shortest possible amount of time.

VIII. CONCLUSION

In this paper we discussed the significance of load balancing in cloud computing and also we discussed various challenges that occur while balancing load in a cloud computing network. Performance of the algorithms which were being implemented for balancing the load in cloud computing was not up to the requirements of cloud. There are different areas involved in achieving the load balancing of a cloud. The problem that we are facing with the existing load balancing algorithms is they are not able to perform well in all the required areas of load balancing.

REFERENCES

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

[2] 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.
[3] wikipedia.org, http://en.wikipedia.org/wiki/Load_balancing_(computing).
[4] V Krishna Reddy, Srikanth Reddy. A Survey of Various Task Scheduling Algorithms in Cloud Computing. i-manager's Journal on Computer Science (JCOM). 2013; 1(1).
[5] http://www-03.ibm.com/press/us/en/pressrelease/22613.wss
[6] Shanti swaroop moharana, Rajadeepan d Ramesh, Digamber powar. Analysis Of Load Balancers In Cloud Computing. International Journal of Computer Science and Engineering (IJCSE). ISSN 2278- 9960. 2013; 2(2).
[7] Martin Randles, Enas Odat, David Lamb, Osama Abu-Rahmeh and A. Taleb-Bendiab,"A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.
[8] S. Dhakal, B. S. Paskaleva, M. M. Hayat, E. Schamiloglu, C. T. Abdallah , Dynamical Discrete-Time Load Balancing in Distributed Systems in the presence of time delays, Decision and Control, 2003. Proceedings. 42nd IEEE Conference, Dec 2003, Volume:5, Pages:5128-5134.
[9] Mequanint Moges, Thomas G.Robertazzi, "Wireless Sensor Networks: Scheduling for Measurement and Data Reporting", August 31, 2005