Vol 6 Issue 5 June 2016

ISSN No : 2230-7850

International Multidisciplinary Research Journal

Indian Streams Research Journal

Welcome to ISRJ

RNI MAHMUL/2011/38595 **ISSN No.2230-7850**

Indian Streams Research Journal is a multidisciplinary research journal, published monthly in English, Hindi & Marathi Language. All research papers submitted to the journal will be double - blind peer reviewed referred by members of the editorial board.Readers will include investigator in universities, research institutes government and industry with research interest in the general subjects.

Address:-Ashok Yakkaldevi 258/34, Raviwar Peth, Solapur - 413 005 Maharashtra, India
Cell : 9595 359 435, Ph No: 02172372010 Email: ayisrj@yahoo.in Website: www.isrj.org

ISSN: 2230-7850 Impact Factor : 4.1625(UIF) Volume - 6 | Issue - 5 | June - 2016

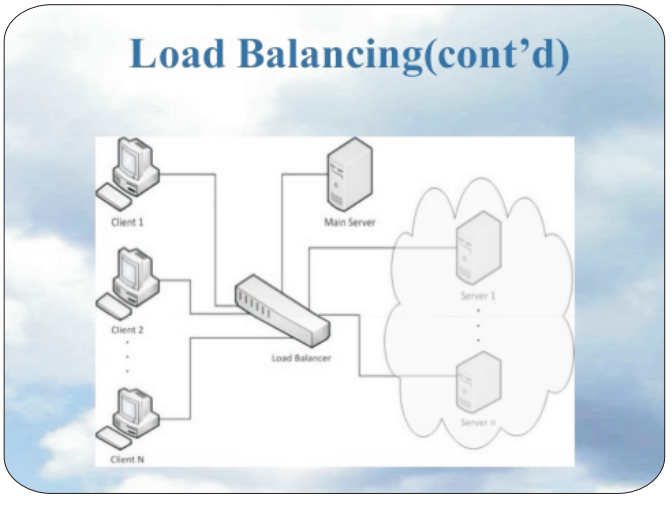
Indian Streams Research Journal

LOAD BALANCING IN CLOUD COMPUTING

Radha Ramani Malladi
Research Scholar, Himalayan University, Arunachal Pradesh, India.

Co - Author Details :

Mahammad. K. Bammanalli
Professor, MESWA's Professional Correspondence College, Opp. Govt. Polytechnic College, VIJAYAPUR (BIJAPUR)



ABSTRACT

Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. Doing so may give rise to certain privacy implications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. It provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications.

Load Balancing is the one of the most important parts of the current virtual environment. The main problem is load balancing in cloud computing and some critical issues like security, privacy, load management and fault tolerance etc. Load balancing helps to distribute all loads between all the nodes. It also ensures that every computing resource is distributed efficiently and fairly. It helps in preventing bottlenecks of the system which may occur due to load imbalance. It provides high satisfaction to the users. Load balancing is a relatively new technique that provides high resource utilization and better response time. There are various advantages of cloud computing including virtual computing environment, on-demand services, maximize resource utilization and easy to use services etc.

In this paper we are providing an overview of cloud technology and its components. We are also focusing on 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.

KEYWORDS :Cloud Computing, Load Balancing, Resource Allocation.

INTRODUCTION

In 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 data center. The distributed computers provide on-demand services. Services may be of software resources (e.g. Software as a Service,SaaS) or physical resources (e.g. Platform as a Service, PaaS) or hardware/infrastructure (e.g. Hardware as a Service, HaaS or Infrastructure as a Service, IaaS).

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

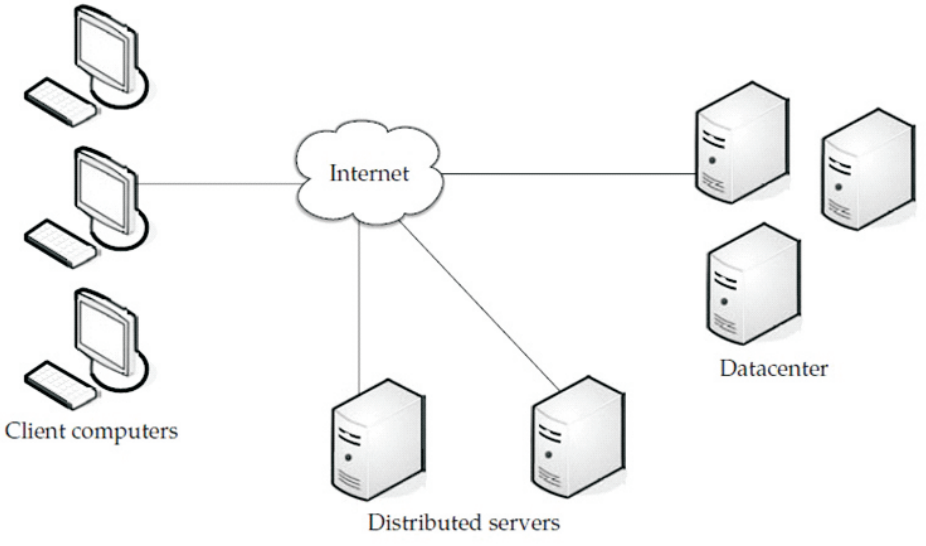


Figure 1: Three components make up a cloud computing solution

Type of Clouds

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

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

Cloud computing is an internet based computing which focuses on the term cloud. Cloud refers to a special network in which large number of host machines or networks are interconnected and which provides shared resources, soft wares and other information. Thus cloud computing refers to accessing this cloud as per user demands.

The characteristics of cloud computing are:

1) User centric
2) Task centric
3) Accessible

Virtualization

It is a very useful concept in context of cloud systems. Virtualisation means" something which isn't real", but gives all the facilities of a real. It is the software implementation of a computer which will execute different programs like a real machine.

Virtualisation is related to cloud, because using virtualisation an end user can use different services of a cloud. The remote data center will provide different services in a fully or partial virtualised manner.

2 types of virtualization are found in case of clouds:

- Full virtualization
- Para virtualization

Full Virtualization

In case of full virtualisation a complete installation of one machine is done on the another machine. It will result in a virtual machine which will have all the soft wares that are present in the actual server.

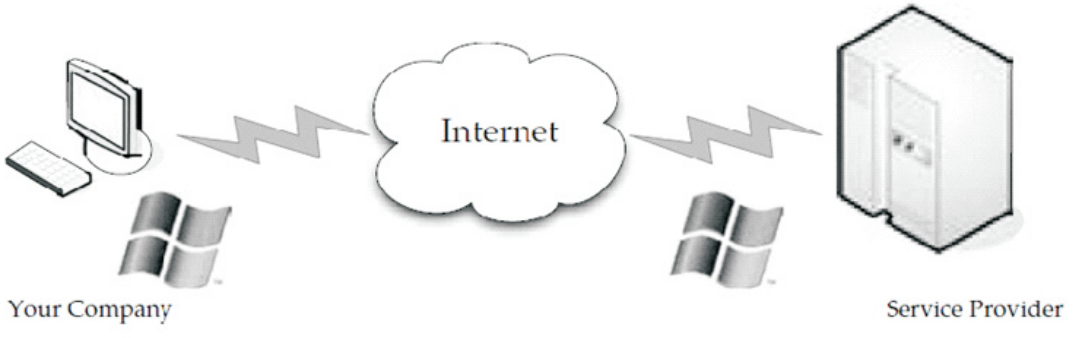


Figure 2: Full Virtualization

Here the remote data center delivers the services in a fully virtualised manner. Full virtualization has been successful for several purposes as:

- Sharing a computer system among multiple users
- Isolating users from each other and from the control program
- Emulating hardware on another machine

PARAVIRTUALIZATION

In para virtualisation, the hardware allows multiple operating systems to run on single machine by efficient use of system resources such as memory and processor. e.g. VM waresoftware. Here all the services are not fully available, rather the services are provided partially.

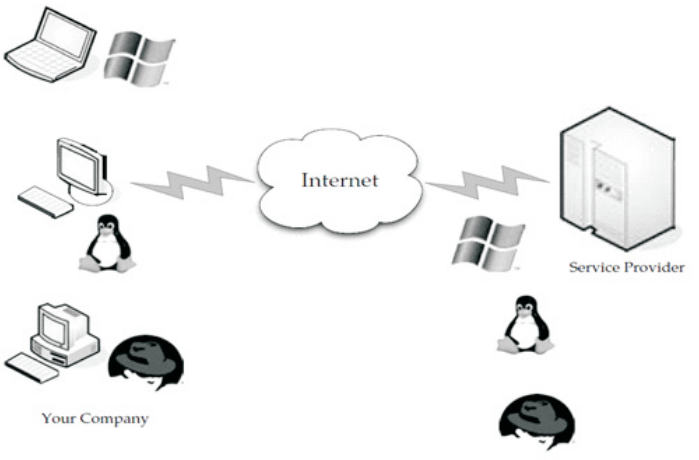


Figure 3: Para virtualization

II. LOAD BALANCING

It is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded.

A load balancing algorithm which is dynamic in nature does not consider the previous state or behaviour of the system, that is, it depends on the present behaviour of the system. The important things to consider while developing such algorithm are: Estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones. This load considered can be in terms of CPU load, amount of memory used, delay or Network load.

GOALS OF LOAD BALANCING

As given, 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

Types of Load balancing algorithms

Depending on who initiated the process, load balancing algorithms can be of three categories:

Sender Initiated: If the load balancing algorithm is initialised by the sender.

Receiver Initiated: If the load balancing algorithm is initiated by the receiver.

Symmetric: It is the combination of both senders initiated and receiver initiated.

Depending on the current state of the system, load balancing algorithms can be divided into 2 categories as:

Static: It doesn't depend on the current state of the system. Prior knowledge of the system is needed.

Dynamic: Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach.

Here we will discuss on various dynamic load balancing algorithms for the clouds of different sizes.

Dynamic Load balancing algorithm

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. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative.

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 independently 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 than the non-distributed ones because, each of the nodes in the system needs to interact with every other node. 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.

Policies or Strategies in dynamic load balancing

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.

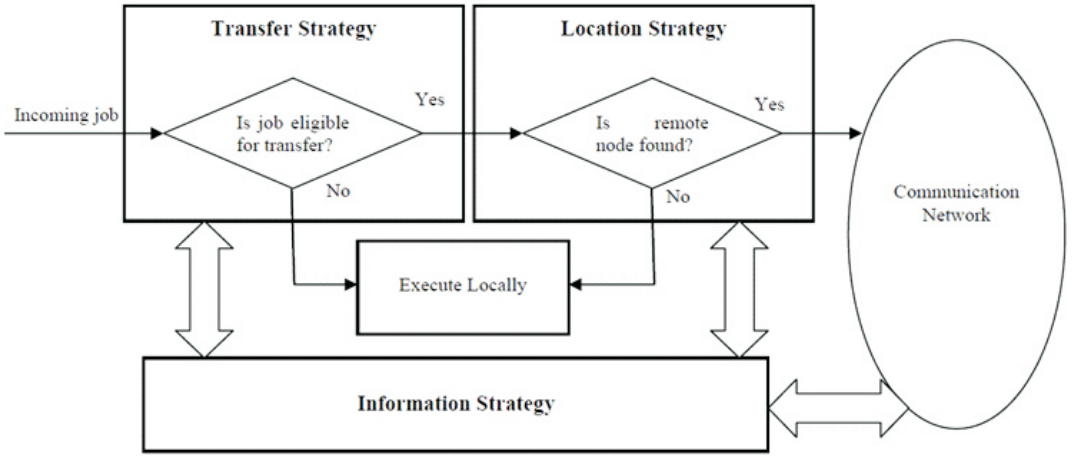


Figure 4: Interaction among components of a dynamic load balancing algorithm

HONEYBEE FORAGING ALGORITHM

This algorithm is derived from the behaviour of honey bees for finding and reaping food. There is a class of bees called the forager bees which forage for food sources, upon finding one, they come back to the beehive to advertise this using a dance called waggle dance. The display of this dance, gives the idea of the quality or quantity of food and also its distance from the beehive. Scout bees then follow the foragers to the location of food and then begin to reap it. They then return to the beehive and do a waggle dance, which gives an idea of how much food is left and hence results in more exploitation or abandonment of the food source.

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 analogousto 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 behaviour, or it can check for advertisements (see dance) and serve it, thus showing scout behaviour.

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 behaviour.

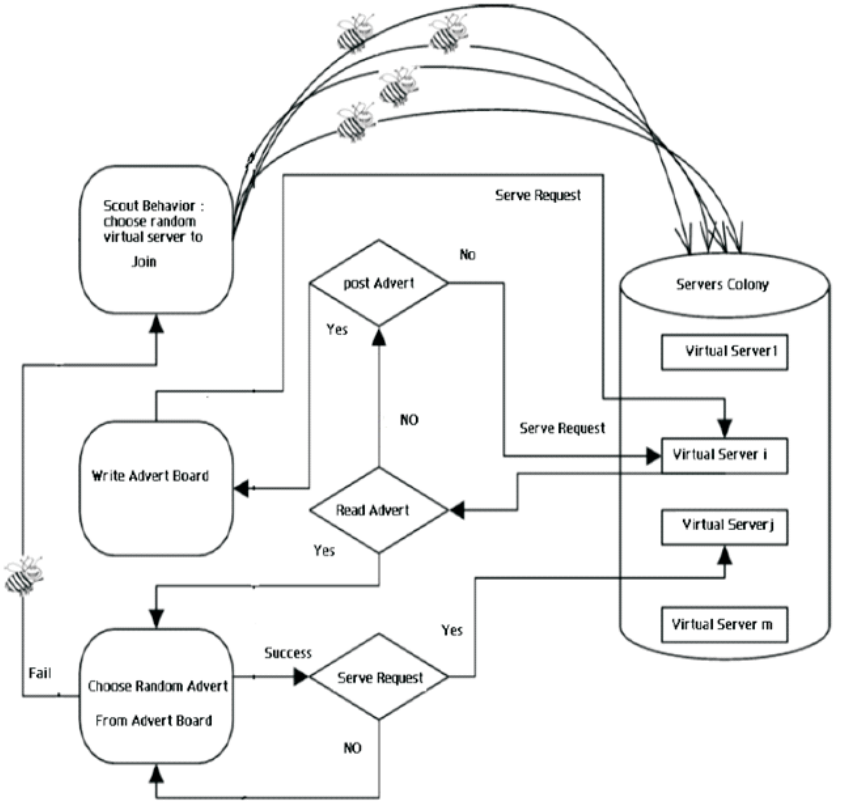


Figure 5: Server Allocations by Foraging in Honey bee technique

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 in degree directed to the free resources of the server.

Regarding job execution and completion, whenever a node does or executes a job, it deletes an incoming edge, which indicates reduction in the availability of free resource.

After completion of a job, the node creates an incoming edge, which indicates an increase in the availability of free resource.

The addition and deletion of processes is done by the process of random sampling. The walk starts at any one node and at every step a neighbour is chosen randomly. The last node is selected for allocation for load. Alternatively, another method can be used for selection of a node for load allocation, that being selecting a node based on certain criteria like computing efficiency, etc. Yet another method can be selecting that node for load allocation which is under loaded i.e. having highest in degree. If b is the walk length, then, as b increases, the efficiency of load allocation increases. We define a threshold value of b, which is generally equal to log n experimentally

A node upon receiving a job, will execute it only if its current walk length is equal to or greater than the threshold value. Else, the walk length of the job under consideration is incremented and another neighbour node is selected randomly. When, a job is executed by a node then in the graph, an incoming edge of that node is deleted. After completion of the job, an edge is created from the node initiating the load allocation process to the node which was executing the job.

Finally, what we get is a directed graph. The load balancing scheme used here is fully

decentralized, thus making it apt for large network systems like that in a cloud.

ACTIVE CLUSTERING

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

A node initiates the process and selects another node called the matchmaker node from its neighbours satisfying the criteria that it should be of a different type than the former one. The so called matchmaker node then forms a connection between a neighbour of it which is of the same type as the initial node.

The matchmaker node then detaches the connection between itself and the initial node. The above set of processes is followed iteratively.

REFERENCES

[1] Marios D. Dikaiakos, George Pall is, Dimitrios Katsaros, Pankaj Mehra, Athena Vakali, 2009 "Cloud computing : Distributed Internet Computing for IT and Scientific Research" IEEE Internet Computing, Published by the IEEE Computer Society.

[2] Peter Mell,Timothy Grance, The NIST Definition of "Cloud Computing" National Institute of Standards and Technology - Computer Security Resource Center-www.csrc.nist.gov.

[3] Cloud Computing, accessed (23/01/2013), from http://en.wikipedia.org/wiki/Cloud_computing.

[4] John Harauz, Lorti M. Kaufinan. Bruce Potter, 2009 "Data Security in the World of Cloud Computing" IEEE Security & Privacy, Copublished by the IEEE Computer and Reliability Societies.

[5] Mohammed A. AlZain, Eric Pardede, Ben Soh, 2012 "Cloud Computing Security: From Single to Multi-Clouds" 45th Hawaii International Conference on System Sciences, IEEE, pp: 7/12.

[6] Cloud Computing Architecture, accessed (27/01/2013), from
http://communication.howstuffworks.com/cloud-computingl.htm

[7] VMware, Cloud computing Virtual infrastructure and virtual machine manager, accessed (25/01/2013), from

[8] Chun-Cheng Lin, Hui-Hsin Chin, Der-Jiunn Deng,"Dynamic Multiservice Load balancing in Cloud-Based Multimedia System", 1932-8184/$31.00 ©2013 IEEE, DOI 10.1109/JSYST.2013.2256320.

[9] Yinchuan Deng, Rynson W.H. Lau, "On DelayAdjustment for Dynamic Load Balancing inDistributed Virtual Environments", IEEETRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 18, NO. 4, APRIL2012

[10] L.D. Dhinesh Babua, P. Venkata Krishna, "Honey beebehavior inspired load balancing of tasks in cloudcomputing environments", SciVerse ScienceDirect'sApplied Soft Computing, ASOC 1894 1–12, © 2013Elsevier B.V.