IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 4, Ver. IV (Jul – Aug. 2014), PP 23-28
www.iosrjournals.org

Dynamic Method for Load Balancing in Cloud Computing

Nikita Haryani[1], Dhanamma Jagli[2]
[1]*Student, Department of MCA, VES Institute of Technology, Mumbai, India*
[2]*Assistant Professor, Department of MCA, VES Institute of Technology, Mumbai, India*

Abstract: *The state-of-art of the technology focuses on data processing and sharing to deal with huge amount of data and client's needs. Cloud computing is a promising technology, which enables one to achieve the aforesaid goal, leading towards enhanced business performance. Cloud computing comes into center of attention immediately when you think about what IT constantly needs: a means to increase capacity or add capabilities on the fly without investing in new infrastructure, training new human resources, or licensing new software. The cloud should provide resources on demand to its clients with high availability, scalability and with reduced cost. Cloud Computing System has widely been adopted by the industry, though there are many existing issues which have not been so far wholly addressed. Load balancing is one of the primary challenges, which is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. This Paper gives an efficient dynamic load balancing algorithm for cloud workload management by which the load can be distributed not only in a balanced approach, but also it allocates the load systematically and uniformly by checking certain parameters like number of requests the server is handling currently. It balances the load on the overloaded node to under loaded node so that response time from the server will decrease and performance of the system is increased.*

I. Introduction

Cloud computing promises to increase the velocity with which applications are deployed, enhance modernization, and lower expenses, all at the same time increasing business agility. Cloud Computing is a concept that has many computers interconnected through a real time network like internet. Cloud computing mainly refers to distributed computing. Cloud computing enables well-situated, on-demand, dynamic and reliable utilization of distributed computing assets. The cloud is altering our life by providing users with new kinds of services. Users acquire service from a cloud without paying attention to the details. Cloud computing is a on demand service in which shared resources work together to perform a task to get the results in minimum possible time by distribution of any dataset among all the connected processing units. Cloud computing is also referred to refer the network based services which give an illusion of providing a real server hardware but in real it is simulated by the software's running on one or more real machines. Such virtual servers do not exist physically so they can be scaled up and down at any point of time [1]. Cloud computing is high utility software having the ability to change the IT software industry and making the software even more attractive [2]. Hence, It helps to accommodate changes in demand and helps any organization in avoiding the capital costs of software and hardware [3] [4].
Cloud computing exhibits several characteristics[5]:

On demand self services: computer services like email, applications, network or server service can be provided with no requirement of human interaction with every service provider. Cloud service providers giving these services on demand self services are Amazon Web Services (AWS), Microsoft, Google, IBM and Salesforce.com.. Gartner describes this quality as service based. New York Times and NASDAQ are examples of companies using AWS (NIST).
Broad network access: Cloud Capabilities are offered over the network and accessed through standard mechanisms that encourage use by mixed thin or thick client platforms such as mobile phones, laptops along with PDAs.
Resource pooling: The provider's computing assets are pooled together to supply multiple clients using multiple-tenant model, with diverse physical and virtual resources dynamically assigned and reassigned according to end user demand. The resources include among others storage space, processing, memory, network bandwidth, virtual machines moreover email services. The pooling collectively of the resource builds economies of scale (Gartner).
Rapid elasticity: Cloud services can be quickly and elastically provisioned, in some cases automatically, to swiftly scale out and rapidly released to quickly scale in. To the consumer, the capabilities accessible for provisioning frequently emerge as unlimited and can be purchased in any quantity at any time.
Measured service: Cloud computing source usage can be measured, controlled, and reported given that transparency for both the provider and consumer of the utilised service. Cloud computing services apply a

metering ability which enables to control and optimise resource use. This means that just similar to air time, electricity or municipality water IT services are charged per usage metrics – pay per use. The additional you utilize the higher the bill. Just as utility companies sell authority or power to subscribers, in addition to telephone companies sell voice and data services, IT services like network security management, data center hosting or yet departmental billing can now be easily delivered as a contractual service.

Multi Tenacity: It is the 6th characteristics of cloud computing advocated by the Cloud Security Alliance. It refer to the need for policy-driven enforcement, segmentation, separation, governance, service levels, as well as chargeback/billing models for different consumer constituencies. Consumers may utilize a public cloud provider's service offerings or actually be from the same organization, like different business units rather than distinct organizational entities, however would still share infrastructure.

There are many problems prevalent in cloud computing [6],[7]. Such as:

- Ensuring appropriate access control (authentication, authorization, as well as auditing)
- Network level migration, so that it requires least cost and time to shift a job
- To offer correct security to the data in transit and to the data at rest.
- Data availability issues in cloud
- official quagmire and transitive trust issues
- Data lineage, data origin and inadvertent leak of sensitive information is possible.

And the most prevalent problem in Cloud computing is the problem of Load Balancing.

II. Necessity Of Load Balancing

Load balancing is a computer network method for distributing workloads across multiple computing resources, for example computers, a computer cluster, network links, central processing units or disk drives. Load balancing plans to optimize resource use, maximize throughput, minimize response time, and evade overload of any one of the resources. By the use of multiple components with load balancing instead of a single component may increase reliability through redundancy.

Load balancing in the cloud differs from classical thinking on load-balancing architecture and implementation by using commodity servers to perform the load balancing because it's difficult to predict the number of requests that will be issued to a server. This provides for new opportunities and economies-of-scale, also presenting its own unique set of challenges. Load balancing is one of the central issues in cloud computing [8]. It is a mechanism that distributes the dynamic local workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. It helps to attain a high customer satisfaction and resource utilization ratio, consequently improving the overall performance and resource utility of the system. It also makes sure that every computing resource is distributed efficiently and fairly [9]. It further prevents bottlenecks of the system which may occur due to load imbalance. When one or more components of any service stop working, load balancing facilitates in continuation of the service by implementing fair-over, i.e. in provisioning and de-provisioning of instances of applications without fail. Fig 1 depicts the Load Balancing necessity in cloud when there are requests from multiple clients. The existing load balancing techniques in clouds, consider various parameters such as performance, response time, scalability, throughput, resource utilization, fault tolerance, migration time and associated overhead. The emerging cloud computing model attempts to address the explosive growth of web-connected devices, and handle massive amounts of data [10] and client demands. Thereby, giving rise to the question whether our cloud model is able to balance the ever-increasing load in an effective way or not.

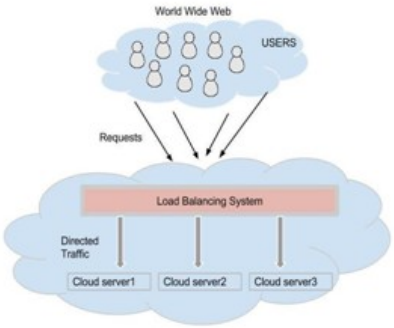

Fig 1. Load Balancing system in cloud computing

III. Literature Review

Some of the chief goals of a load balancing algorithm, as pointed out by [11] are:

- To accomplish a greater overall progress in system performance at a realistic cost, e.g., decrease task response time while keeping acceptable delays;
- To treat all jobs in the system equally not considering their origin;
- To encompass a fault tolerance: performance survival under partial failure in the system;
- To have the ability to alter itself in accordance with any changes;
- preserve system stability

The important things to consider while developing such algorithm are : 1)estimation of load 2)comparison of load 3)stability of different systems 4)performance of system 5)interaction between the nodes 6)nature of work to be transferred 7) selecting of nodes and many other ones . This load considered can be in terms of CPU load, amount of memory used, delay or else Network load.

We can divide Load balancing algorithms into 2 categories, Depending on the state of the system.

- **Static:** It doesn't depend on the current state of the system. Prior knowledge of the system is essential.
- **Dynamic:** Decisions on load balancing are based on current state of the system. No prior knowledge is required. Thus it is better than static approach.

In a distributed system, dynamic load balancing can be done in two different ways [12]:

- Distributed
- non-distributed

A dynamic load balancing algorithm assumes no previous knowledge about job actions or the global state of the system, i.e., load balancing decisions is exclusively based on the existing or current status of the system. 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 realize load balancing can take two forms: 1) cooperative and 2) non-cooperative. In the cooperative, the nodes work side-by-side to attain a common goal, for example, to advance the overall response time, etc. In the non-cooperative, every node works independently in the direction of a goal local to it, for example, to advance the response time of a local task. Dynamic load balancing algorithms having distributed nature, frequently produce more messages than the non-distributed ones because, each of the nodes in the system is required to interact with every other node. The advantage, of this is that even if one or more nodes in the arrangement fail, it will not cause the total load balancing process to stop; it instead would influence the system performance to a little extent.
In non-distributed type, either one node or a group of nodes perform the task of load balancing. Dynamic load balancing algorithms of non-distributed nature can get two forms: 1) centralized and 2) semi-distributed. In the centralized, the load balancing algorithm is executed just by a single node in the total system: the central node. This node is exclusively in charge for load balancing of the whole system. The other nodes interact merely with the central node. However, in semi-distributed form, nodes are partitioned into clusters, where the load balancing in every cluster is of centralized form. A central node is chosen in each cluster by suitable election technique which takes care of load balancing inside that cluster. Hence, the load balancing of the complete system is done via the central nodes of each cluster. Centralized dynamic load balancing takes less messages to arrive at a decision, since the number of overall interactions in the system decreases drastically as compared to the semi-distributed case. However, centralized algorithms can create a bottleneck in the system at the central node and also the load balancing process is rendered hopeless once the central node crashes. Therefore, this algorithm is mainly suited for networks with small size. Thus, fig 2 shows the summarization of dynamic load balancing technique.

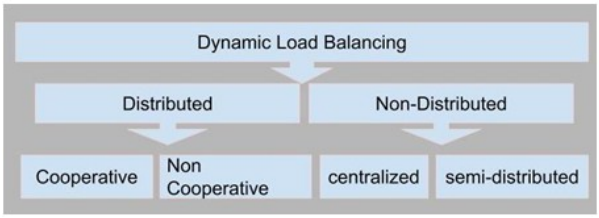

Fig 2: Summarizing Dynamic load balancing techniques

Round Robin: In this algorithm [13], the processes are divided between all processors. Each process is handed over to the processor in a round robin order. The process allotment order is maintained in the vicinity independent of the allotments from remote processors. However the work load distributions between processors are the same but the job processing time for dissimilar processes are not same. So by any point of time some nodes may be greatly loaded and others wait at leisure. This algorithm is frequently used in web servers where http requests are of alike nature and scattered likewise.

Connection Mechanism: Load balancing algorithm can as well be based on least connection mechanism which is a component of dynamic scheduling algorithm. It requires to count the number of connections for each server dynamically to approximate the load. The load balancer keeps track of the connection number of each server. The number of link adds to when a new connection is sent out to it, and decreases the number when connection terminate or timeout happens [14].

A Task Scheduling Algorithm Based on Load Balancing: This is discussed in [15] a two-level task scheduling method based on load balancing to convene dynamic requirements of users and obtain high resource utilization. It accomplishes load balancing by first mapping tasks to virtual machines and then virtual machines to host resources by this means improving the task response time, resource consumption and overall performance of the cloud computing environment.

Randomized: Randomized algorithm is of type static in nature [16]. In this algorithm a process can be handled by a particular node n with a probability p. The process allocation order is preserved for each processor independent of allotment from remote processor. This algorithm facilitates well in case of processes that are equal loaded. On the other hand, trouble arises when loads are of different computational complexities. Randomized algorithm does not keep up deterministic approach. It facilitates well while Round Robin algorithm generates overhead for process queue.

IV. Proposed System

A dynamic load balancing algorithm makes load distribution decisions based on the current work load at every node of the distributed system. Accordingly, this algorithm must provide a means for collecting and managing system status information.

The algorithm handles the requests in a proficient way. It starts by checking the counter variable of each server node and data center. After checking, it transfers the load accordingly by choosing the minimum value of the counter variable and the request is handled easily and takes a smaller amount of time, and offers maximum throughput. The randomly transfer of load can cause some server to heavily loaded while other server is lightly loaded. If the load is equally distributed it not only improves performance also reduces the time delay. This algorithm not only balances the load but also it improves the response time for the cloud. While taking into account the impact of cost optimization one has to think on the subject of the solution to this difficulty.

A counter variable is related with each node. Counter variable is the number of requests that the particular server node or data center is currently handling. Each node is having multiple data centers as shown in fig 3. The value of counter variable of server node will be equal to the sum of counter variables of its data centers.

This algorithm essentially allocates request which is coming from the client nodes to the lightly loaded server cluster (Data Center) and gives the response in a reduced amount of time by doing this, it makes the algorithm proficient for response to request ratio. We can see that the clients at a same time make requests to access the cloud application over the internet.

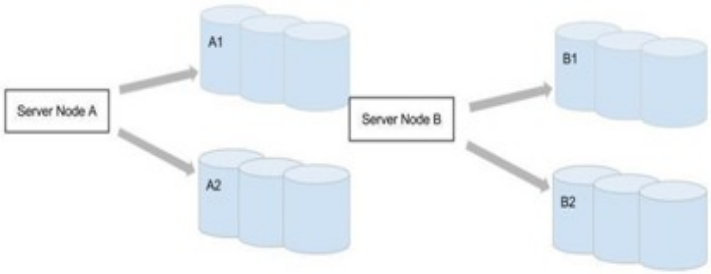

Fig 3: Server Nodes having multiple data centers

Algorithm:

1) Select the node with minimum value of counter variable (i.e. minimum number of requests allotted)
2) Assign the application request to the selected node.
3) Select the data center (of the selected node in step 1) that has minimum value of counter variable.

4) Assign the client request to the selected data center
5) Increase the counter variable of the data center by 1
6) Execute the application request
7) Decrease the value of counter variable by 1

Now in this algorithm all the requests goes through the load balancer system as shown in fig 2 by which it checks the counter variable which is associated with each server node set to the maximum requests currently handled by a server node. Here cntA, cntB, cntA1, cntA2, cntB1, cntB2 are the counter variables of server node A, server node B, data center A1, data center A2, data center B1, data center B2 respectively as shown in table 1. Let us assume that the node A is having a counter value to 120, node B is having the counter variable to 115. Node B is handling the smaller number of requests compared to node A, so here the load balancer will stabilize the load (requests) to node B as it is less hence the balancing is done at this level. Now, deciding which server cluster (data center) of node B will handle the request. Suppose in node B, cluster B1 is having counter variable set to 70, cluster B2 to 45. As cluster B2 is handling smaller number of requests compared to other cluster of node B, so the request will be allotted to cluster B2 in order to balance the overall load and counter variable associated with cluster B2 will be incremented by 1. And also counter variable associated with server node B will be incremented by 1 i.e. now it will become 116 as shown in table 2. Figure 4 shows the workflow of the algorithm. Till now we have handled the request however how the counter variable will get updated? The answer is the servers which the counter variable is associated with, will simultaneously change (update) the counter variable. When a response is given back to the client the server will automatically decrease its counter variable by the number 1 and also the counter variable of related cluster will be decremented by 1, so that every time the algorithm will have the updated value of counter variable. Therefore, requests are handled easily by Server Clusters. The potency of server can be increased or decreased by the service provider on request and for data centers too. So no requirement of Round Robin Balancing or any other practice where time is consumed and response to request ratio is small for huge number of requests.

Component	Counter Variable	value
Server Node A	cntA	120
Data Center A1	cntA1	65
Data Center A2	cntA2	55
Server Node B	cntB	115
Data Center B1	cntB1	70
Data Center B2	cntB2	45

Table 1: Initial state of the system

Component	Counter Variable	value
Server Node A	cntA	120
Data Center A1	cntA1	65
Data Center A2	cntA2	55
Server Node B	cntB	116
Data Center B1	cntB1	70
Data Center B2	cntB2	46

Table 2: State of the system after assigning client request

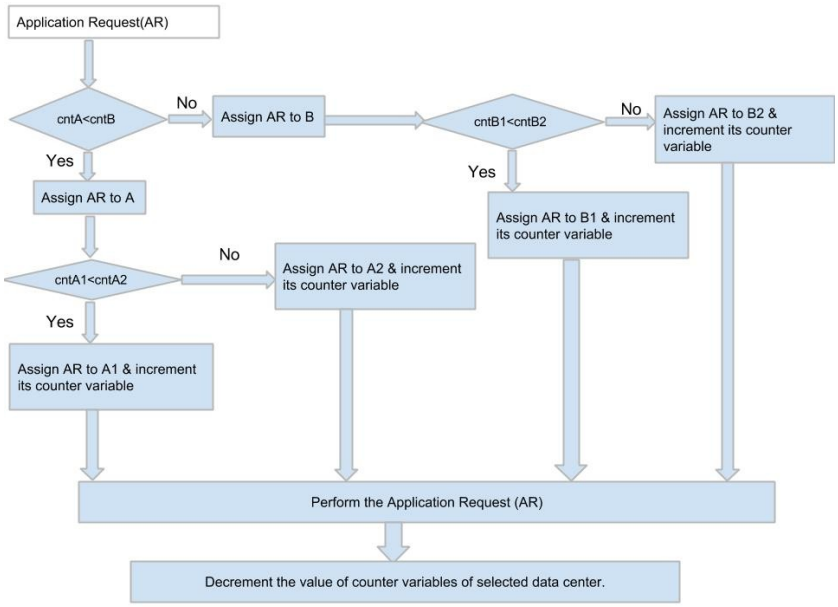

Fig 4: Workflow of the algorithm

V. Conclusion

Existing Load Balancing techniques/Algorithms that have been considered largely focus on reducing overhead, reducing the migration time and improving performance etc., but the response to request ratio is rarely considered. It is a challenge of every engineer to build up the cloud platforms that can raise the throughput. In proposed algorithm, the request is allotted as early as possible to the appropriate data center. As there are various server nodes having multiple data centers, the response is given at the earliest, thereby distributing the load in a balanced and efficient manner without any delay. Because of the dynamic nature of the algorithm, there is no need to have the prior knowledge of the state of the system; hence the overhead for storing the previous state of the system is also eliminated.

References

[1]. S.Hemachander Harikrishna, R.Backiyalakshmi, "A Game Theory Modal Based On Cloud Computing For Public Cloud", IOSR Journal of Computer Engineering, Volume 16, Issue 2, Ver. XII (Mar-Apr. 2014).

[2]. 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 Vol. 2, Issue 2, May 2013, 101-108.

[3]. R. W. Lucky, "Cloud computing", IEEE Journal of Spectrum, Vol. 46, No. 5, May 2009, pages 27-45.

[4]. M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Journal of Internet Computing, Vol. 13, No. 5, September/October 2009, pages 10-13.

[5]. http://www.isaca.org/groups/professional-english/cloud-computing/groupdocuments/essential%20characteristics%20of%20cloud%20computing.pdf

[6]. T.R.V. Anandharajan, Dr. M.A. Bhagyaveni" Co-operative Scheduled Energy Aware Load-Balancing technique for an Efficient Computational Cloud" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[7]. Wayne Jansen Timothy Grance" Guidelines on Security and Privacy in Public Cloud Computing" NIST Draft Special Publication 800-144.

[8]. 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

[9]. A. M. Alakeel, "A Guide to dynamic Load balancing in Distributed Computer Systems", International Journal of Computer Science and Network Security (IJCSNS), Vol. 10, No. 6, June 2010, pages 153-160.

[10]. C.Kishor Kumar Reddy, P.R Anisha, K.Srinivasulu Reddy, S.Surender Reddy, IOSR Journal of Computer Engineering (IOSRJCE) ,ISSN: 2278-0661 Volume 2, Issue 1 (July-Aug. 2012), PP 39-46

[11]. http://www.ukessays.com/essays/information-technology/implementating-distributed-load-balancing-algorithms-information-technology-essay.php

[12]. 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.

[13]. Zhong Xu, Rong Huang,(2009)"Performance Study of Load Balanacing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[14]. P.Warstein, H.Situ and Z.Huang(2010), "Load balancing in a cluster computer" In proceeding of the seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE.

[15]. Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277

[16]. Zhong Xu, Rong Huang,(2009)"Performance Study of Load Balanacing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.