An Systematic Overview On Cloud Computing and Load Balancing in the Cloud

Neha Gohar Khan
P.R.Patil College of Engg & Technology, Amravati

Prof. V. B. Bhagat
P.R.Patil College of Engg & Technology, Amravati

Abstract:

Cloud Computing (or cloud for short) is a compelling technology. Cloud computing is emerging as a new paradigm of large-scale distributed computing. Load balancing is one of the main challenges in cloud computing which is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. This paper consists the concepts of cloud computing and some of the methods of load balancing in large scale Cloud systems. Our aim is to provide an evaluation and comparative study of these approaches, demonstrating different algorithms for load balancing and to improve the different performance parameters like throughput,response time,latency etc. for the clouds.

Keywords: Cloud computing**,**Load balancing, static algorithms, dynamic algorithms,metrics for load balancing.

INTRODUCTION:

In 1969, Leonard Kleinrock , one of the chief scientists of the original Advanced Research Projects Agency Network (ARPANET) which seeded the Internet, said: “As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, I will probably see the spread of “computer utilities” which, like present electric and telephone utilities, will service individual homes and offices across the country”.[1]Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage[5].It’s a term which is generally used in case of Internet. The whole Internet can be viewed as a cloud. Users get service from a cloud without paying attention to the details. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. It provides the scalable IT resources such as applications and services, as well as the infrastructure on which they operate, over the Internet, on pay-per-use basis to adjust the capacity quickly and easily[8].It helps to accommodate changes in demand and helps any organization in avoiding the capital costs of software and hardware.[4]The idea of cloud computing is based on a very fundamental principal of reusability of IT capabilities. The difference that cloud computing brings compared to traditional concepts of “grid computing”, “distributed computing”,“utility computing”, or “autonomic computing” is to broaden horizons across organizational boundaries. Forrester defines cloud computing as:*“A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption”.*Thus,cloud computing is a framework for enabling a suitable on-demand network access to a shared pool of computing resources (e.g. networks, servers, storage, applications,and services).

Cloud Components

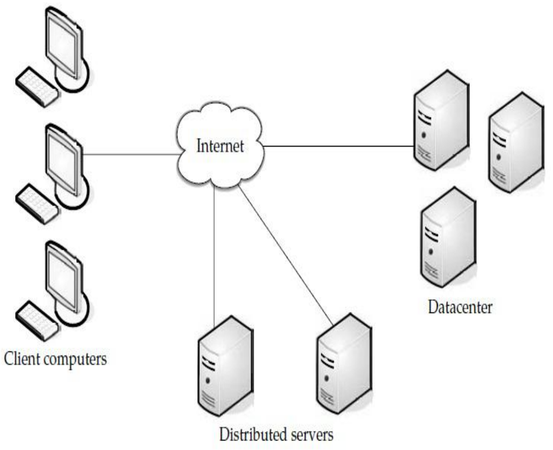


Fig: Three components make up a cloud computing solution

A Cloud system consists of three major components such as clients, datacenter, and distributed servers. Each element has a definite purpose and plays a specific role[11].

Load Balancing in Cloud Computing

Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers. Load balancing in the cloud computing environment has an important impact on the performance. Good load balancing makes cloud computing more efficient and improves user satisfaction.

Load balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes, making sure that no single node is overwhelmed, hence improving the overall performance of the system[4]. Load balancing is a relatively new technique that facilitates networks and resources by providing a maximum throughput with minimum response time. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. Dividing the traffic between servers,data can be sent and receivedwithout major delay. Different kinds of algorithms are available that helps traffic loaded between available servers . A basic example of load balancing in our daily life can be related to websites. Without load balancing, users could experience delays, timeouts and possible long systemresponses.[2]This paper focuses mainly on the prevalent load balancing techniques in cloud computing environment.

Load Balancing is done with the help of load balancers where each incoming request is redirected and is transparent to client who makes the request. Based on predetermined parameters, such as availability or current load, the load balancer uses various scheduling algorithm to determine which server should handle and forwards the request on to the selected server. To make the final determination, the load balancer retrieves information about the candidate server’s health and current workload in order to verify its ability to respond to that request.

Load balancer features:

Load balancers may have a variety of special features such as:

Priority activation: When the number of available servers drops below a certain number, or load gets too high, standby servers can be brought online.

Distributed Denial of Service (DDoS) attack protection: load balancers can provide features such as SYN cookies and delayed-binding (the back-end servers don't see the client until it finishes its TCP handshake) to mitigate SYN flood attacks and generally offload work from the servers to a more efficient platform.

Health checking: the balancer polls servers for application layer health and removes failed servers from the pool.

TCP buffering: the load balancer can buffer responses from the server and spoon-feed the data out to slow clients, allowing the web server to free a thread for other tasks faster than it would if it had to send the entire request to the client directly.

HTTP caching: the balancer stores static content so that some requests can be handled without contacting the servers.

Content filtering: some balancers can arbitrarily modify traffic on the way through.

Client authentication: authenticate users against a variety of authentication sources before allowing them access to a website.

Firewall: direct connections to backend servers are prevented, for network security reasons Firewall is a set of rules that decide whether the traffic may pass through an interface or not.

Intrusion prevention system: offer application layer security in addition to network/transport layer offered by firewall security.

Existing load balancing algorithms for cloud computing:

1.Weighted Active Monitoring load balancing algorithm

The 'Weighted Active Monitoring Load Algorithm' is implemented; modifying the Active Monitoring Load Balancer by assigning a weight to each VM as discussed in Weighted Round Robin Algorithm of cloud computing in order to achieve better response time and processing time.

2. Dynamic Load Balancing Algorithms:

The three methods are:

- **SA -** Simulated Annealing: We directly minimize the above cost function by a process analogous to slow physical cooling.
- **ORB -** Orthogonal Recursive Bisection: A simple method which cuts the graph into two by a vertical cut, then cuts each half into two by a horizontal cut, then each quarter is cut vertically, and so on.
- **ERB -** Eigenvector Recursive Bisection: This method also cuts the graph in two then each half into two, and so on, but the cutting is done using an eigenvector of a matrix with the same sparsity structure as the adjacency matrix of the graph. The method is an approximation to a computational neural net.

Static algorithms

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more imperfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic .

Dynamic algorithms

Dynamic algorithms designated proper weights 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.In comparisonbetween these two algorithms,although round robin algorithms based on simple rule, more loads conceived on servers and thus imbalanced traffic discovered as a result [2].

Distribute workload of multiple network links to achieve maximum throughput, minimize response time and to avoid overloading. We describe three algorithms to distribute the load and check the performance time and cost.[1]

A. Round Robin Algorithm (RR):

It is the simplest algorithm that uses the concept of time quantum or slices Here the time is divided into multiple slices and each node is given a particular time quantum or time interval and in this quantum the node will perform its operations. The resources of the service provider are provided to the client on the basis of this time quantum. In Round Robin Scheduling the time quantum play a very important role for scheduling, because if time quantum is very large then Round Robin Scheduling Algorithm is same as the FCFS Scheduling. If the time quantum is extremely too small then Round Robin Scheduling is called as Processor Sharing Algorithm and number of context switches is very high. It selects the load on random basis and leads to the situation where some nodes are heavily loaded and some are lightly loaded. Though the algorithm is very simple but there is an additional load on the scheduler to decide the size of quantum[5] and it has longer average waiting time, higher context switches, higher turnaround time and low throughput.

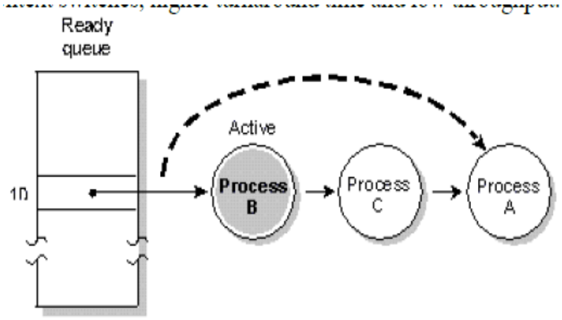


Figure 1. Round Robin Algorithm

B. Equally Spread Current Execution Algorithm (ESCE):

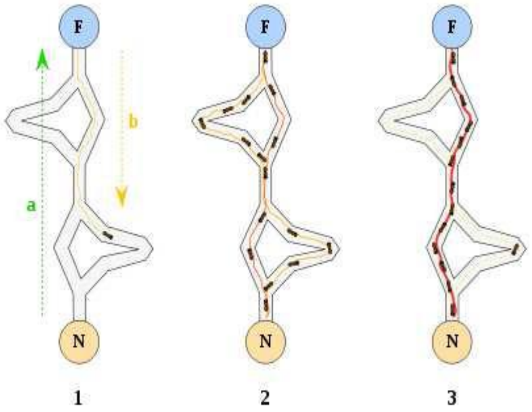
In spread spectrum technique load balancer makes effort to preserve equal load to all the virtual machines connected with the data centre. Load balancer maintains an index table of Virtual machines as well as number of requests currently assigned to the Virtual Machine (VM). If the request comes from the data centre to allocate the new VM, it scans the index table for least loaded VM. In case there are more than one VM is found than first identified VM is selected for handling the request of the client/node, the load balancer also returns the VM id to the data centre controller. The data centre communicates the request to the VM identified by that id. The data centre revises the index table by increasing the allocation count of identified VM. When VM completes the assigned task, a request is communicated to data centre which is further notified by the load balancer. The load balancer again revises the index table by decreasing the allocation count for identified VM by one but there is an additional computation overhead to scan the queue again and again

C. Throttled Load Balancing Algorithm (TLB):

In this algorithm the load balancer maintains an index table of virtual machines as well as their states (Available or Busy). The client/server first makes a request to data centre to find a suitable virtual machine (VM) to perform the recommended job. The data centre queries the load balancer for allocation of the VM. The load balancer scans the index table from top until the first available VM is found or the index table is scanned fully. If the VM is found, the load data centre. The data centre communicates the request to the VM identified by the id. Further, the data centre acknowledges the load balancer of the new allocation and the data centre revises the index table accordingly. While processing the request of client, if appropriate VM is not found, the load balancer returns -1 to the data centre. The data centre queues the request with it. When the VM completes the allocated task, a request is acknowledged to data centre, which is further apprised to load balancer to de- allocate the same VM whose id is already communicated. The total execution time is estimated in three phases. In the first phase the formation of the virtual machines and they will be idle waiting for the scheduler to schedule the jobs in the queue, once jobs are allocated, the virtual machines in the cloud will start processing, which is the second phase, and finally in the third phase the cleanup or the destruction of the virtual machines. The throughput of the computing model can be estimated as the total number of jobs executed within a time span without considering the virtual machine formation time and destruction time The proposed algorithm will improve the performance by providing the resources on demand, resulting in increased number of job executions and thus reducing the rejection in the number of jobs submitted.[1]

D.Ant colony optimization algorithms

Genetic Algorithms (GA) have been used to evolve computer programs for specific tasks, and to design other computational structures. The recent resurgence of interest in AP with GA has been spurred by the work on Genetic Programming (GP).[2]GP paradigm provides a way to do program induction by searching the space of possible computer programs for an individual computer program that is highly fit in solving or approximately solving the problem at hand.[9]The geneticprogramming paradigm permits the evolution of computer programs which can perform alternative computations conditioned on the outcome of intermediate calculations, which can perform computations on variables of many different types, which can perform iterations and recursions to achieve the desired result, which can define and subsequently use computed values and subprograms, and whose size, shape, and complexity is not specified in advance. GP use relatively low-level primitives, which are defined separately rather than combined a priori into high-level primitives, since such mechanism generate hierarchical structures that would facilitate the creation of new high-level primitives from built-in low-level primitives.



1. The first ant finds the food source (F), via any way (a), then returns to the nest (N), leaving behind a trail pheromone (b)
2. Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route.
3. Ants take the shortest route; long portions of other ways lose their trail pheromones.[2]

Metrics For Load Balancing In Clouds

The existing load balancing techniques in clouds, consider various parameters like performance, response time, scalability, throughput, resource utilization, fault tolerance, migration time and associated overhead.Some of them are discussed below:

Throughput- is used to calculate the no. of tasks whose exe-cution has been completed. It should be high to improve the performance of the system.

Response Time- is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.

Resource Utilization- is used to check the utilization of re-sources. It should be optimized for an efficient load balancing.

Scalability- is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.

Performance-is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

Overhead Associated-determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.

Fault Tolerance- is the ability of an algorithm to perform uni-form load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.

Migration time- is the time to migrate the jobs or resources from one node to other. It should be minimized in order to enhance the performance of the system[4].

CONCLUSION

Cloud Computing has widely been adopted by the industry, though there are many existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation, Energy Management, etc. which have not been fully addressed. Central to these issues is the issue of load balancing, that is required to distribute the excess dynamic local workload evenly to all the nodes in the whole Cloud to achieve a high user satisfaction and resource utilization ratio by ensuring that every computing resource is distributed efficiently and fairly. The motivation of the survey of existing load balancing techniques in cloud computing is to encourage the amateur researcher to contribute in developing more efficient load balancing algorithms. This information might be useful for interested researchers to carry out further work in this research area.Our future study and research includes the strategies for load balancing using cloud partitioning for the public cloud.

REFERENCES:

[1]Tejinder Sharma, Vijay Kumar Banga "Efficient and Enhanced Algorithm in Cloud Computing",International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-1, March 2013.

[2]Doddini Probhuling L.,"Load balancing algorithms in cloud computing",International Journal of Advanced Computer and Mathematical Sciences ISSN 2230-9624. Vol4, Issue3, 2013, pp229-233 http://bipublication.com.

[3]A Load Balancing Model Based on Cloud Partitioning for the Public Cloud, IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013.

[4] Nidhi Jain Kansal "Cloud Load Balancing Techniques : A Step Towards GreenComputing", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012 ISSN (Online): 1694-0814www.IJCSI.org.

[5] Nidhi Jain Kansal* and Inderveer Chana, "Existing load balancing techniques in cloud computing: A systematic re-view", journal of information systems and communication issn: 0976-8742, e-issn: 0976-8750, volume 3, issue 1, 2012, pp- 87-91.http://www.bioinfo.in/contents.php?id=45.

[6] Mishra , Ratan , Jaiswal, Anant,P"Ant Colony Optimiza tion: A Solution Of Load Balancing In Cloud",April 2012, International Journal Of Web & Semantic Technology;Apr2012, Vol. 3 Issue 2, P33.

[7] Eddy Caron, Luis Rodero-Merino "Auto-Scaling, Load Balancing And Monitoring In Commercial And Open-Source Cloud" ResearchReport ,January2012.

[8] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia,April 2010, pages 551-556.

[9] 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.

[10] B. Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale. com/info center/whitepapers/ Load-Balancing-in-the-Cloud.pdf, 2012.

[11]Ram Prasad Padhy,P Goutam Prasad Rao, "LOAD BALANCING IN CLOUD COMPUTING SYSTEMS", Department of Computer Science and Engineering National Institute of Technology, Rourkela,Orissa, India.pdf

[12]http://www.livinginternet.com/w/wionline.htm.

[13] www.cloudbus.org/cloudsim.

[14]http://www.ca.com/~/media/Files/whitepapers/turnkey_clouds_turnkey_profits.pdf.