A Review of Existing Load Balancing Techniques in Cloud Computing

Amritpal Singh

Abstract— Cloud computing paradigm is rapidly transforming the manner in which we avail the various services and resources over the Internet load balancing is a method which allocates workload across different nodes to ensure that none of the node is overwhelmed or is lacking resources. Load balancing algorithms in cloud computing environment allocate workload across different nodes to attain optimum resource utilization, enhanced throughput, speedy response time avoiding bottlenecks etc. Load balancing algorithms are chiefly classified under static and dynamic algorithms there are various algorithms under these two classifications like Honey Bee Foraging Algorithm, Throttled load Balancing Algorithm, Ant Colony Optimization Algorithm etc. for evaluation of the algorithms there are different performance metrics used like throughput, resource utilization, scalability, performance etc. this paper will review the different types of load balancing algorithms which are utilized in cloud computing environment

Index Terms— Cloud Computing, load Balancing, Workload, Honey Bee Foraging, Performance Metrics

I INTRODUCTION

Cloud computing has redefined the way we access services and resources over the Internet with cloud approach services and resources ranging from software, applications, network bandwidth etc. can be availed over the Internet all these services and resources are managed by cloud service providers cloud computing offers three service models infrastructure as a service (IaaS), software as a service (SaaS), platform as a service (PaaS). Cloud computing is a model for enabling ubiquitous, convenient on demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [2]. Cloud computing facility can deliver services in form of software (e.g. email, web browser), platform (e.g. development tools) and infrastructure (e.g. storage space) it is a service oriented application many firms are relying on cloud computing paradigm to cater to the needs of users it has its own share of benefits and challenges and has tremendous scope for Future [1]

Fig 1 a cloud computing model

II Load Balancing

Load balancing is one of the paramount tasks undertaken to enhance the performance of a cloud and to attain optimal resource utilization, it is an inherent part of managing a cloud computing environment load balancing techniques strive to attain enhanced throughput, shortened response time, avoiding congestion, quick response time etc. Load may comprise of amount of memory used, CPU load, network load, delay load etc. the aim of load balancing methods is to distribute the load evenly among different nodes so that no node gets overwhelmed. 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. [4]. As cloud service providers dispense different services and resources through servers load balancing methods ensure efficient working of the servers by taking suitable action if any server is overloaded or if some is in an idle state due to lack of resources.

III Goals of Load Balancing

1. **Optimum utilization of resources**: it is one of the paramount aims of load balancing as proper resource utilization is essential for efficiency of the cloud model.

2. **High Throughput**: high throughput is a desired attribute required for a high performance system which is

ISSN: 2278 – 1323

only feasible if the workload and resources are dispensed to the different nodes evenly

3. **Short response time**: Load balancing methods strive to

4. **Avoiding bottlenecks**: To avoid any type of congestion or bottlenecks in the cloud environment either in network or data availability.

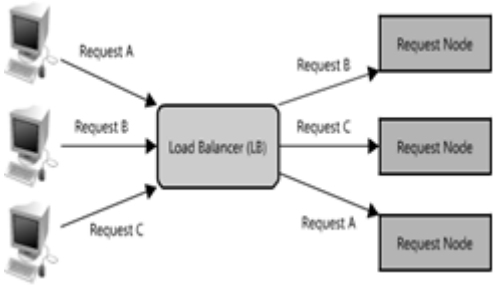

Fig 2 Load balancing in cloud computing

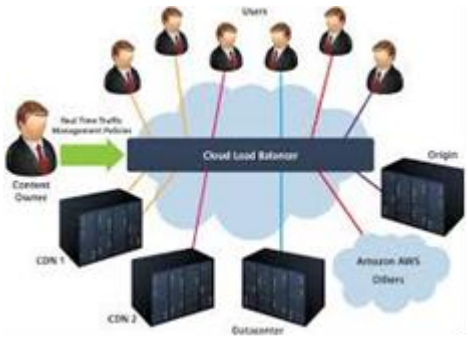

Fig 3 Load balancing in cloud computing

IV Performance metrics for load balancing algorithms

1. **Throughput**: in general terms it is defined as the amount of work completed in a specified period of time high throughput is sought for higher performance of a system

2. **Response time**: in a distributed environment the time it takes for a load balancing algorithm to start responding to a given instruction is defined as response time for a better performance a short response time is desired.

3. **Fault tolerance**: it is the ability of a load balancing algorithm to continue its operation if some error occurs a high fault tolerance is desired for optimal system performance

4. **Scalability**: it is the capability of the load balancing algorithm to able to manage a growing amount of workload and increasing number of nodes as the cloud expands.

5. **Resource utilization**: it is the factor which describes the extent to which a resource is utilized for efficient performance high resource utilization is desired.

6. **Forecasting accuracy**: it is the degree of conformity of the output that is obtained to the one which was expected after calculation

V Types of Load balancing Algorithm

1. **Static algorithms**: the execution of these algorithms do not take into account the current state of the system hence these algorithms don't depend on the current state which the system is in, static algorithms have prior knowledge related to system resources and details of all tasks. These algorithms allocate workload among processors before execution of algorithm Depending on their performance such as arrival time execution time, amount of resources needed ,the workload is distributed in the start by master processor the slave processor calculate its allocated work and submits the result to the master the goal of SLB method is to reduce the overall execution time of a concurrent program and minimizing the communication delays [3]. These are easy to design and implement but are inapt to face a sudden failure e.g. Round robin Algorithm, Randomized Algorithm, Central Manager Algorithm.

2. **Dynamic algorithms**: dynamic algorithms take the current system state into account while taking decisions unlike static algorithms which don't do so these algorithms don't require any prior knowledge of the system state, workload is distributed among the processors during the execution of the algorithm unlike the static algorithm dynamic algorithm buffers the process in the queue on the main node and allocates dynamically upon request from remote nodes as a result, dynamic load balancing algorithm can provide a significant improvement in performance over static algorithm [3] these algorithms are complex but more fault tolerant and have an overall better performance than static algorithms e.g. Parallel Graph Partitioning.

3. **Sender initiated algorithm**: this type of load balancing algorithm is initiated by the sender here the sender transmits the request messages until it gets a receiver that can accept the workload

4. **Receiver initiated algorithm**: this type of load balancing algorithm is initiated by the receiver here the receiver transmits the request messages till it finds a sender that can accept the workload

5. **Symmetric algorithm**: this algorithm is a combination of sender initiated and receiver initiated algorithm

6. **Centralized approach**: in centralized approach the tasks and workload is passed from a centralized location to different processes there is a master slave relation between the centralized location and processes

7. **Decentralized approach**: in this approach the workload is passed to arbitrary processes

ISSN: 2278 – 1323

VI Existing load balancing Algorithms

1. **Honey Bee Foraging Algorithm**: This algorithm emulates the behavior of honey bees to find food so its named honey bee foraging algorithm this algorithm is derived from a detailed analysis of the behavior that honey bees adopt to find and reap food. In bee hives, there is a class of bees called the scout bees which forage for food sources, upon finding one, they come back to the beehive to advertise this using a dance called waggle/tremble/vibration dance the display of this dance, gives the idea of the quality and/or quantity of food and also its distance from the beehive. Forager bees then follow the Scout Bees to the location of food and then begin to reap it they then return to the beehive and do a waggle or tremble or vibration dance to other bees in the hive giving an idea of how much food is left and hence resulting in either more exploitation or abandonment of the food source [5]

Based on these phenomena of bees searching for food the algorithm works in the same manner, the removed tasks from over loaded VMs are considered as the honey bees upon submission to the under loaded VM, the task will update the number of various priority tasks and load of that particular VM to all other waiting tasks this will be helpful for other tasks in choosing their virtual machine based on load and priorities. Whenever a high priority task has to be sub mitted to other VMs, it should consider the VM that has minimum number of high priority tasks so that the particular task will be executed at the earliest. Since all VMs will be sorted in ascending order based on load, the task removed will be submitted to under loaded VM. In essence, the tasks are the honey bees and the VMs are the food sources [5].

2. **CARTON**: R Stanojevic et al. [6] proposed this mechanism it works by unifying the use of LB (Load Balancing) and DRL (distributed Rate Limited) , load balancing is used to allocate equally the different tasks/jobs to various servers DRL ensures that the resources are distributed in a manner to maintain an equitable and fair allocation of resources DRL also adjusts to the capacity of servers for dynamic workloads to ensure similar level of performance at all servers the main factor tackled by this algorithm is usage control without infinite bandwidth assumption the algorithm is simple, easy to implement and has very low communication and computation overhead.

3. **CLBVM**: A Bhadani et al. [7] proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) it balances the load evenly in a distributed virtual machine/cloud computing environment this approach takes into account that network load is constant and does not change frequently, each virtual machine (VM) has a different identification, in this algorithm a load-balancing policy with a central dispatcher called Central Load Balancing Policy for Virtual Machines (CLBVM) is proposed which makes load-balancing decisions based on global state information. This policy has centralized information and location rules the transfer rule is partially distributed and partially centralized in this algorithm the systems were made completely distributed so that if the performance the VM gets affected by another VM it can move itself to a lightly loaded server on the fly this implementation is strictly meant for Xen based systems

4**. Server based Load Balancing for Internet distributed services**: A M Nakai et al. [8] proposed a load balancing algorithm which is server based and caters to the needs of servers distributed all over the world this algorithm reduces the service response times by utilizing a protocol which redirects requests to the closest remote servers without overloading them, a middleware is described that implements this protocol and it has a heuristic based on the protocol that tolerates abrupt load changes the authors implemented a simulator based on realistic Internet models and real Internet latencies

5. **Biased Random Sampling**: M Randles et al. [9] proposed an approach which depicts load on the server by its connectivity in a virtual graph the nodes of the graph depict the server nodes the in-degree of graph is mapped to the servers free resources or some other measure of desirability this approach creates a network system that provides a measure of initial availability status and as it evolves gives job allocation and usage dynamics when a node executes a new job, it removes an incoming edge decreasing its in-degree and indicating available resources are reduced. Conversely, when the node completes a job, it follows a process to create a new inward edge indicating available resources are increased again the increment and decrement process is performed via random sampling the sampling walk starts at a specific node; at each step moving to a neighbour node chosen randomly it is a scalable technique.

6. **Active Clustering**: M Randles et al. [9] investigated a self-aggregation load balancing algorithm it groups like (i.e. Similar service type) instances together it consists of iterative execution by each node in the network at a random point a node becomes initiator and chooses a match maker node randomly from its current neighbours ,the match maker makes a link between one of the match makers neighbours that is similar to initiator node finally match maker removes the link between itself and the initiator node the main aim of grouping similar nodes is to optimize job assignments by connecting similar services.

7. **Compare and Balance**: Yi Zhao et al. [10] proposed an algorithm addressing the problem of intra- cloud load balancing among physical hosts by adaptive live migration of virtual machines this method is based on Sampling to attain equilibrium the algorithm converges to equilibrium very fast it decreases the migration time of the virtual machines (VM) by shared storage fulfills the zero-downtime relocation of virtual machines by transforming them as Red Hat cluster services.

8. **Event Driven approach**: V Nea et al. [11] proposed a load balancing algorithm for real time Massively Multiplayer Online Games (MMOG) it is a cost-efficient hosting of MMOG sessions on Cloud resources, provisioned

ISSN: 2278 – 1323

on-demand in the correct amount based on the current number of connected players. The resource allocation is driven by a load balancing algorithm that appropriately distributes the load such that the QoS requirements are fulfilled at all times this algorithm after receiving capacity events from capacity planning service as inputs analyses its components in context of the resources and the global state of the game session directs the resource allocation service in taking the appropriate measures for load balancing

9. **Ant Colony and Complex Network Theory Load Balancing (ACCLB):** Z Zhang et al. [12] proposed a load balancing algorithm based on ant colony and complex network theory (ACCLB) in an open cloud federation this method brings into utility small world and scale free characteristics of a complex network to achieve enhanced load balancing this method overcomes heterogeneity it adapts to dynamic environments, has good fault tolerance and is very stable and enhances the overall performance of the system

VII Conclusion

Cloud computing has revolutionized the way resources and services are availed by users over the Internet but it has its challenges efficient load balancing is one of the key issues concerning any cloud service provider as an even distribution of workload across different nodes is a pivotal requirement for high resource utilization and user satisfaction there are different classifications of load balancing algorithms each algorithm gives optimum results in a particular circumstance and scenario, depending on objectives of the cloud environment and given resources an algorithm is selected. The performance of the load balancing algorithms is evaluated by different parameters like throughput, response time, fault tolerance, scalability etc. load balancing is the requirement of a cloud environment and how well this requirement is met depends on the algorithm chosen.

REFERENCES

[1] Amritpal singh "An Overview of Cloud Computing" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 7, July 2015

[2] Pragati Priyadarshinee, Pragya Jain "Load Balancing and Parallelism in Cloud Computing" International Journal of Engineering and Advanced Technology (IJEAT) Volume-1, Issue-5, June 2012

[3] Rajesh George Rajan, V.Jeyakrishnan "A Survey on Load Balancing in Cloud Computing Environments" International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 12, December 2013

[4] Nidhi Jain Kansal, Inderveer Chana "Cloud Load Balancing Techniques: A Step Towards Green Computing" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012

[5] Dhinesh Babu L.D, P. VenkataKrishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied Soft Computing 13 (2013) 2292–2303.

[6] R. Stanojevic, and R. Shorten, "Load balancing vs. distributed rate limiting: a unifying framework for cloud control", Proceedings of IEEE ICC, Dresden, Germany, August 2009, pages 1-6

[7] A. Bhadani, and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE), January 2010.

[8] A. M. Nakai, E. Madeira, and L. E. Buzato, "Load Balancing for Internet Distributed Services Using Limited Redirection Rates",5th IEEE Latin-American Symposium on Dependable Computing (LADC), 2011, pages 156-165.

[9] 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.

[10] Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea, August 2009, pages 170-175.

[11] Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea ,August 2009, pages 170-175.

[12] 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.

[13] Abhijit A. Rajguru, S.S. Apte "A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters" International Journal of Recent Technology and Engineering (IJRTE) Volume-1, Issue-3, August 2012

[14] Zenon Chaczko et al. "Availability and Load Balancing in Cloud Computing" 2011 International Conference on Computer and Software Modeling IPCSIT vol.14 (2011) © (2011) IACSIT Press, Singapore

ISSN: 2278 – 1323