A Survey of Load Balancing Techniques in Cloud Computing

Namrata Swarnkar [1], Asst. Prof. Atesh Kumar Singh [2], Dr. R. Shankar [3]

IES Institute of Technology and Management,
Bhopal (Madhya Pradesh), India

Abstract

Load Balancing is essential for efficient operations in distributed environments. As Cloud Computing is one of the greatest platform which provides storage of data in very lower cost and available for all time over the internet, load balancing for the cloud has become a very interesting and important research area. Load balancing helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Many algorithms were suggested to provide efficient mechanisms and algorithms to enhance the overall performance of the Cloud and provide the user more satisfying and efficient service. In this paper, we investigate the different algorithms proposed to resolve the issue of load balancing in cloud computing.

Keywords— Cloud Computing, Load Balancing, Cloud Service Models, Task Scheduling, virtualization.

I. INTRODUCTION

Cloud Computing:

"Cloud computing" is a term, which 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, and flexibility, reduced overhead for users, reduced cost of ownership, on demand services. In case of 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 but according to the NIST definition of 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, server, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1].

Cloud Computing Architecture

Abstract Layers:

Cloud computing architecture has necessary abstract layers beginning at bottom and working upwards. Figure 1 illustrates the five layers that constitute cloud computing. The bottom layer is the physical hardware (Haas). Customers who use this layer of the cloud are usually big corporations who require an extremely large amount of subleased Hardware as a Service. As a result, the cloud-provider runs, oversees, and upgrades its subleased hardware for its customers [2] .

The next layer consists of the cloud's software kernel. This layer acts as a bridge between the data processing performed in the cloud's hardware layer and the software infrastructure layer which operates the hardware. It is the Lowest level of abstraction implemented by the cloud's software and its main job is to manage the server's hardware resources while at the same time allowing other programs to run and utilize these same resources [3].

The abstraction layer above the software kernel is called software infrastructure. This layer renders basic network resources to the two layers above it in order to facilitate new cloud software environments and applications that can be delivered to end-users in the form of IT services. The services offered in the software infrastructure layer can be separated into three different subcategories: computational resources (IaaS), data storage, and communication [3].

Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications [3].

Data-Storage as a Service (DaaS), allows users of a cloud to store their data on servers located in remote locations and have instant access to their information from any site that has an Internet connection [2][3]. This technology allows software platforms and applications to extend beyond the physical servers on which they reside.

Communication as a Service (CaaS): CaaS to perform services like network security, real-time adjustment of virtual overlays to provide better networking bandwidth

or traffic flow, and network monitoring. Through network monitoring, cloud-providers can track the portion of network resources being used by each customer [2].

Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider [3][4].

Software as a Service (SaaS): The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure2. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface [3][4].

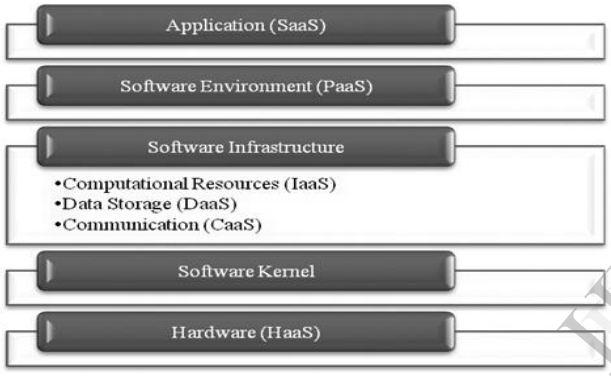

Figure: 1 Cloud Computing Architecture

II. CLOUD VIRTULIZATION

It is a very useful concept in context of cloud systems. Virtualization 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. Virtualization is related to cloud, because using virtualization an end user can use different services of a cloud. The remote datacenter will provide different services in a full or partial virtualized manner [5].Two types of virtualization are found in case of clouds as given in

- **Full Virtualization**
  Full virtualization a complete installation of one machine is done on another machine. It will result in a virtual machine which will have all the software that is present in the actual server.
  Full virtualization has been successful for several purposes:-

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

- **Para-virtualization**
  In Para-virtualization, the hardware allows multiple operating systems to run on single machine by efficient use of system resources such as memory and processor. e.g. VMware software. Here all the services are not fully available, rather the services are provided partially [6].

1. **Disaster recovery**: In the event of a system failure, guest instances are moved to hardware until the machine is repaired or replaced.
2. **Migration**: As the hardware can be replaced easily, hence migrating or moving the different parts of a new machine is faster and easier.
3. **Capacity management**: In a virtualized environment, it is easier and faster to add more hard drive capacity and processing power. As the system parts or hardware can be moved or replaced or repaired easily, capacity management is simple and easier [6].

III LOAD BALANCING

Load balancing is one of the central issues in cloud computing. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work [7]. 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 [8].

Need of Load Balancing in Cloud Computing

Load balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio, making sure that no single node is overwhelmed, hence improving the overall performance of the system. 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 [8].

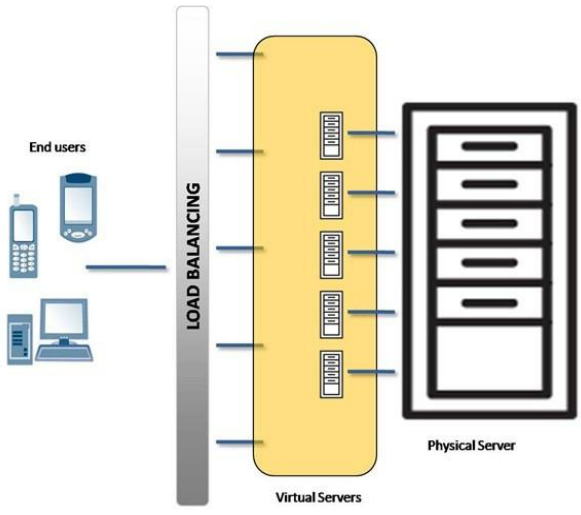

Figure 2: Load Balancing in Cloud Computing

IV OVERVIEW OF LOAD BALANCING ALGORITHMS

1. Round Robin Algorithm: Round Robin is a very famous load balancing algorithm, in which the processes are divided between all processors. The process allocation order is maintained locally independent of the allocations from remote processors. In Round Robin, it send the requests to the node with the least number of connections, so at any point of time some node may be heavily loaded and other remain idle [5], this problem is reduced by CLBDM.

2. Central Load Balancing Decision Model (CLBDM): CLBDM is a central load balancing decision model, which is suggested by Radojevic and Mario Zagar [9], it's based on session switching at the application layer. The improvement is that, in the cloud it calculated the connection time between the client and the node, and if that connection time exceeds a threshold then connection will be terminated and task will be forwarded to another node using the regular Round Robin rules.

2. MapReduce-based Entity Resolution: MapReduce is a computing model and an associated implementation for processing and generating large datasets [10]. Map task and reduce task two main task in this model which written by the user, Map takes an input pair and produces a set of intermediate value pair and Reduce task accepts an intermediate key and a set of values for that key and merges these values to form a smaller set of value. Map task read entities in parallel and process them, this will cause the Reduce task to be overloaded.

3. Ant colony optimization (ACO): Kumar Nishant suggested an algorithm [11] of ant colony optimization. In ACO [11] algorithm when the request in initiated the ant start its movement. Movement of ant is of two ways*: Forward Movement* : Forward Movement means the ant in continuously moving from one overloaded node to another node and check it is overloaded or under loaded ,if ant find an over loaded node it will continuously moving in the forward direction and check each nodes .*Backward Movement*: If an ant find an over loaded node the ant will use the back ward movement to get to the previous node, in the algorithm [11] if ant finds the target node then ant will commit suicide, this algorithm reduced the unnecessary back ward movement ,overcome heterogeneity, is excellent in fault tolerance.

4. Load balancing of virtual machine resources: J. Hu et al. [12] proposed a scheduling strategy on load balancing of VM resources that uses historical data and current state of the system. This strategy achieves the best load balancing and reduced dynamic migration by using a genetic algorithm. It helps in resolving the issue of load-imbalance and high cost of migration thus achieving better resource utilization.

5. Index Name Server Algorithm (INS): The INS algorithm proposed in [13] the goal is to find an algorithm to minimize the data duplication and redundancy. INS is able to handle the load balancing dynamically .INS have some parameters which help in calculating the optimum selection point like that Hash Code of the block of data to be downloaded, the position of the server, the transition quality, the maximum bandwidth. Another calculation point whether the connection can handle additional nodes or not. They classified the busy levels B(a),B(b), and B(c). B(a) means that connection is very busy and cannot handle any additional connection. B(b) means connections is not busy and can handle additional connections. B(c) means that the connection is limited.

6. *Opportunistic Load Balancing (OLB):* Sang proposed OLB is a static load balancing algorithm that has the goal of keeping each node in cloud busy [14]. However OLB does not calculate the execution time of the node, due to this the tasks to be processed in a slower manner and will cause bottlenecks since requests might be pending waiting for nodes to be free.

7. *Load Balancing Min-Min Algorithm (LBMM):* Wang suggested an algorithm called LBMM [15]. LBMM has a three level load balancing framework. In first level LBMM architecture is the request manager which is responsible for receiving the task and assigning it to service manager, when the service manager receives the request; it divides it into subtask and assigns the subtask to a service node based on node availability, remaining memory and the transmission rate which is responsible for execution the task.

8. Dual Direction Downloading Algorithm (DDFTP): DDFTP is a dual direction downloading algorithm from FTP server [16]. This algorithm can be also implemented for Cloud Computing load balancing. This is a fast and efficient concurrent technique for downloading large files from FTP server in a cloud environment. DDFTP uses the concept of processing the

files for transfer from two different directions. For example, one server will start from block 0 and keeps downloading incrementally while another server start from block m and keeps downloading in a decrement order. When the two servers download two consecutive blocks, the task is considered as finished and other task can be assigned to the server. As a result, both servers will work independently. The algorithm reduces the network communication between the client and nodes and network overhead.

9. Exponential Smooth Forecast-based on Weighted Lest Connection (ESBWLC): The algorithm proposed in [17] is a dynamic load balancing algorithm for cloud computing. ESBWLC build the conclusion of assigning a certain task to a node after having a number of task assigned to that service node and getting to know the node's CPU power, memory, number of connections and the amount of disk space currently in used, then ESBWLC predicts which node is to be selected based on exponential smoothing.

10. A Lock-free multiprocessing solution for LB - X. Liu et al. [18] proposed a lock-free multiprocessing load balancing solution that avoids the use of shared memory in contrast to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. It is achieved by modifying Linux kernel. This solution helps in improving the overall performance of load balancer in a multi-core environment by running multiple load-balancing processes in one load balancer.

11. *Honeybee Foraging Behavior* - M. Randles et al. [19] investigated a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased system diversity but throughput is not increased with an increase in system size. It is best suited for the conditions where the diverse population of service types is required.

V CHALLENGES IN CLOUD COMPUTING LOAD BALANCING

A. *Overhead Associated* - determines the amount of overhead involved while implementing a load-balancing algorithm [20]. 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.

B. *Throughput* - is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system [20].

C. *Performance* – is used to check the efficiency of the system. It has to be improved at a reasonable cost e.g. reduce response time while keeping acceptable delays [20].

D. *Resource Utilization* - is used to check the utilization of resources. It should be optimized for an efficient load balancing [21].

E. *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 [21].

F. *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 [21].

G. *Fault Tolerance* - is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique [7].

H. *Point of Failure*: Control the load balancing and collect data of different nodes and the system designed in a way that avoids the single point of failure in the algorithms. For example centralized algorithms, if one controller fails, then the whole system would fail. Any load balancing algorithm must be designed in order to overcome this challenge [22].

VI CONCLUSION

In this paper, we surveyed multiple algorithms and discussed the need of Load balancing in cloud computing and metrics for load balancing in cloud. We also discussed the Cloud Virtualization. In cloud computing load balancing is the main issue. Load balancing is required to distribute the excess dynamic local workload evenly to the entire node in the whole cloud to achieve a high user satisfaction and resource utilization ratio. It also ensures that every computing resource is distributed efficiently and fairly.

References

[1] National institute of standards and Technology computer security Resource Center –www.CSRC.nist.gov

[2] *Fei Hu, Meikang Qiu, Jiayin li, Travis Grant, Draw Tylor,Seth McCaleb, Lee Butler and Richard Hamner,"A Review on Cloud Computing:Design Challenges in Architecture and Security" journal of Computing and Information Technology-CIT 19, 2011.*

[3] *LizheWang,Jie Tao, Marcel Kunze"Scientific Cloud Computing: Early Definition and Experience"The 10th IEEE International Confrrence Computing and Communications 2008.*

[4] *Software & Information Industry Association, "Softwaor as a Service: Strategic Backgrounder", February 2001.*

[5] *Sotomayor, B., RS. Montero, IM. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," in IEEE Internet Computing, Vol. 13, No. 5, pp: 14-22, 2009*

[6] Ni, J., Y. Huang, Z. Luan, J. Zhang and D. Qian, "Virtual machine mapping policy based on load balancing in private cloud environment," in proc. International Conference on Cloud and Service Computing (CSC), IEEE, pp: 292-295, December 2011.

[7] Rimal, B. Prasad, E. Choi and I. Lumb, "A taxonomy and survey of cloud computing systems." In proc. 5th International Joint Conference on INC, IMS and IDC, IEEE, 2009.

[8] Pradeep K.Sinha, "Distributed operating Systems Concepts and Design" IEEE Computer Society Press.

[9] Radojevic, B. and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." In proc.34th International Convention on MIPRO, IEEE, 2011.

[10] Kolb, L., A. Thor, and E. Rahm, E, "Load Balancing for MapReduce-based Entity Resolution," in proc. 28th International Conference on Data Engineering (ICDE), IEEE, pp: 618-629, 2012

[11] Nishant, K. P. Sharma, V. Krishna, C. Gupta, KP. Singh, N. Nitin and R. Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization." In proc. 14th International Conference on Computer Modelling and Simulation (UKSim), IEEE, pp: 3-8, March 2012.

[12] J. Hu, J. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010.

[13] T-Y., W-T. Lee, Y-S. Lin, Y-S. Lin, H-L. Chan and J-S. Huang, "Dynamic load balancing mechanism based on cloud storage" in proc. Computing, Communications and Applications Conference (ComComAp), IEEE, January 2012.

[14] Sang, A., X. Wang, M. Madihian and RD. Gitlin, "Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems," in Wireless Networks, Vol. 14, No. 1, pp: 103-120, January 2008.

[15] Wang, S-C., K-Q. Yan, W-P. Liao and S-S. Wang, "Towards a load balancing in a three-level cloud computing network," in proc. 3rd International Conference on. Computer Science and Information Technology (ICCSIT), IEEE, Vol. 1, pp: 108-113, July 2010

[16] Al-Jaroodi, J. and N. Mohamed. "DDFTP: Dual-Direction FTP," in proc. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp:504-503, May 2011.

[17] Ren, X., R. Lin and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast" in proc. International Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE, pp: 220-224, September 2011.

[18] Xi. Liu, Lei. Pan, Chong-jun. Wang, and Jun-Yuan. Xie, "A Lock-free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent System and Application (ISA), 2011, pages 1-4.

[19]] 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.

[20] Foster, I., Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-degree compared," in proc. Grid Computing Environments Workshop, pp: 99-106, 2008.

[21] Buyya R., R. Ranjan and RN. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in proc. 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Busan, South Korea, 2010.

[22] Ranjan, R., L. Zhao, X. Wu, A. Liu, A. Quiroz and M. Parashar, "Peer-to-peer cloud provisioning: Service discovery and load-balancing," in Cloud Computing - Principles, Systems and Applications, pp: 195-217, 2010.