Effective Distributed Dynamic Load Balancing for the Clouds

Swapnil Jaykar Mohite[1], Abhishek Suresh Natekar [2], Rajat Sanjay Lohar[3], Mr. S. R. Kadam[4]
[1,2,3,4] *UG Student Nanasaheb Mahadik College of Engineering, Peth, Department of Computer Science and Engineering, NMCOE College of Engineering, DBATU University, Lonere, India*

Abstract **- "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, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Central to these issues lies the establishment of an effective load balancing algorithm. 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. 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.**

Index Terms **- Cloud Computing, Load Balancing, CaCo Algorithm.**

I.INTRODUCTION

Distributed file system is the basic building block of cloud computing. In distributed file system, a large file is scattered into number of chunks and allocates each chunk to separate node to perform processing function parallel over each unique node. In cloud, if number of storage nodes and number of files are increase and assesses to that file is also increases then the central node becomes obstacle. The resource scheduling task is used to reduce (eliminate) the load on central node. With the help of load scheduling algorithm, the load of nodes is balanced as well as the movement cost is also minimized. It results in load asymmetry in distributed filing system. to overcome the load asymmetry/imbalance drawback, a completely distributed Load scheduling algorithmic has been applied, that is active in nature doesn't cogitate the previous state or behavior of the system and it completely depends on this behavior of the system and approximation of system load, performance of system, comparison of load, stability of various system, interaction between the nodes, nature of load to be moved, choice of nodes and network traffic.

II. OBJECTIVES

- To Implement a system in Cloud environment.
- To impellent dynamic job ordering with the help of Genetic Algorithm
- To design an approach to each task can be provide to perfect resource as allocation.
- To improve the efficiency and reduce the time complexity of system.

III. RELATED WORK

Dey,et.al, [1] their paper presents an effective as well as efficient load balancing strategy to address these hurdles effectively: a secure heuristic load balancing approach for cloud data centers based on RSA and SFQ. The effective balance of incoming loads in cloud data centers is their approach. With the help of Start Fair Queuing (SFQ) algorithm, they are able to the fair allocation of VMs in their proposed approach. Hence, it maximizes the resource utilization. With the help of RSA algorithm, they are able to take care of security concerns of their proposed approach.

Shailendra Narayan Singh, et.al, [2] aim of the author's paper was the scheduling of non-preemptive jobs is an irretrievable limit and therefore been allocated to the suitable virtual machines in the beginning placement the aforementioned. Almost, the onset tasks incorporate several reliant jobs and they may complete self-governing jobs in several virtual machines and in the same virtual machines several cores. Moreover, tasks occur in the course of server execution in fluctuating arbitrary interims in several

load situations. The contributing varied resources are accomplished by distributing the jobs to suitable assets by dynamic and static scheduling to build the cloud more effective and thus it advances the customer fulfillment. The aim of this study advocates the load balancing and scheduling procedure by keeping in mind the competences of virtual machines, the job dimension of every demanded task or interdependency of several tasks.

Arpitha Raghunandan**,** et.al, [3] says that a way to design this system for adaptive video streaming. This system is highly scalable and can handle high loads, i.e., a higher number of users connecting to the application simultaneously. This paper proposes an algorithm called inter-server load balancing algorithm with Adaptive Agent-based load balancing to solve this problem. The algorithms also incorporate dynamic video resolution delivery techniques to ensure smooth viewing experience in the whole user experience irrespective of the network speed and bandwidth.

Xuejiang, et.al, [4] says that a new sensor cloud architecture for IoT based on fog computing, which preprocesses raw sensor data on fog node and provides temporary storage of the preprocessing results if it is needed, controls and manages diverse type of sensors in IoT devices through the virtualization of physical sensors, and ultimately provides dynamic, on-demand, elastic and standardized Sensing-as-a-Service to end users. They also design the architecture and the components of fog-based sensor cloud system, give a detail function view of the components, and put forward the preliminary solutions for physical sensor virtualization, dynamic provisioning of virtual sensor group and service instance.

Mohammad, et al, [5] In their paper they mentioned a joint computing-plus-communication optimization framework exploiting virtualization technologies, called MMGreen. Authors proposal specifically addresses the typical scenario of multimedia data processing with computationally intensive tasks and exchange of a big volume of data. The proposed framework not only ensures users the Quality of Service (through Service Level Agreements), but also achieves maximum energy saving and attains green cloud computing goals in a fully distributed fashion by utilizing the DVFS-based CPU frequencies. To evaluate the actual effectiveness of the proposed framework, they conduct experiments with MMGreen under real-world and synthetic workload traces. The results of the experiments show that MMGreen may significantly reduce the energy cost for computing, communication and reconfiguration with respect to the previous resource provisioning strategies, respecting the SLA constraints.

Litian, et al, [6] In their paper they mentioned an entropy weight clustering scheduling (EWCS) algorithm, which combines the dynamic heterogeneous redundancy (DHR) architecture of mimetic defense theory and K-Means clustering of machine learning to complete the nodes selection on the cloud platform. This algorithm consists of four steps: risk value screening, load balancing, entropy weight calculation and clustering optimization. The simulation results show that the algorithm is reasonable and can serve MCOE well. It is also an effective attempt to apply machine learning method to scheduling problem.

IV. METHODOLOGY

The design goals for Constraint Scheduler were:

1) To be able to give users immediate feedback on whether the job can be completed within the given deadline or not and proceed with execution if deadline can be met. Otherwise, users have the option to resubmit with modified deadline requirements.

2) Maximize the number of jobs that can be run in the cluster while satisfying the time requirements of all jobs depicts how our scheduler components fit into the cloud architecture.

Assuming that 15 slots are available to these three users in the global (logical) slot table, Alice will be allocated 8 slots, Bob 3 slots and Sam 4 slots. Exactly how these slots are mapped to physical nodes is not guaranteed. Whenever a slot becomes available the allocations are recalculated to determine who should get the new slot according to their granted share. Furthermore, local tasks are attempted first. If that fails, remote rack tasks are scheduled. There may be opportunities to delay scheduling of some jobs to achieve a higher ratio of data local tasks. However, in the current implementation we enforce the shares strictly in each time period. Packing a user on a single node versus distributing the job workload across nodes is another application specific trade-off that we may address in future implementations.

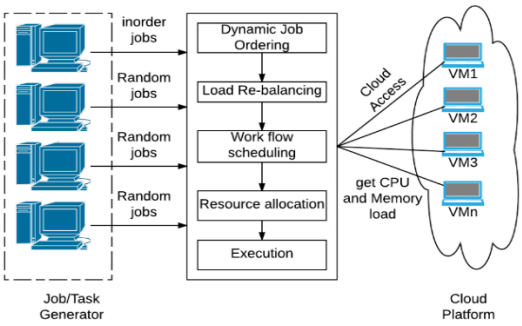

Fig. Proposed System Architecture

The proposed system overall execution. The system first creates the multiple random jobs and assigns to middle ware system. The system is in control for dynamically completely jobs ordering as well as each virtual machine load balancing. The algorithm has used for optimize the all VM's. Scheduling and resource allocation has done after the ordering and scheduling. The system also focus on dynamic data recover approach using matrix generation. First system create some data nodes with some matrix and for every data file, and when disk failure has generate, it will recover all original data from available data nodes and matrix.

- Energy cost constitutes a significant portion of operation cost
- for cloud system.
- Efficient storage for ant type of data hosting in cloud environment.
- Tackle energy cost minimization problem
- Inefficient utilization of resources over cloud
- Improve response time

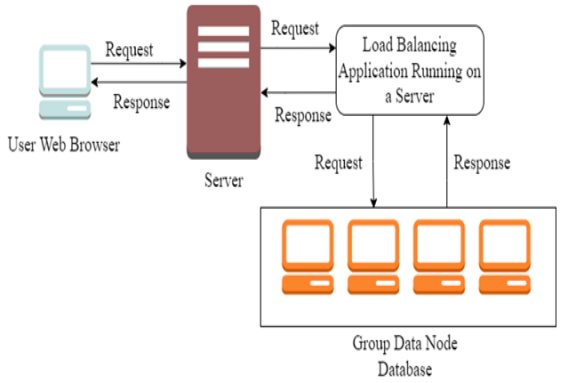

Fig. Conceptual Design

The proposed work focus how multi authentication system can work in multi cloud environment. This is the first model in cloud framework which is providing multiple verification entities. Basically, the system can provide more effective results when works with multi cloud environment, it can overcome some policy base encryption methods as well as foreword security approaches.

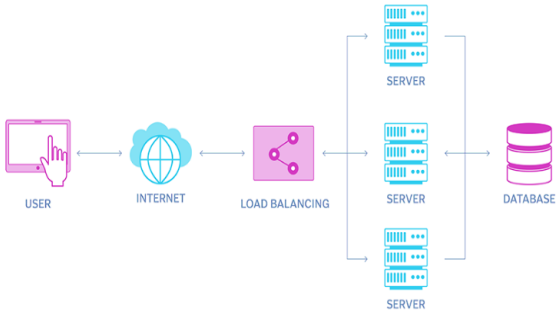

Fig. Working of Load Balancing

Enterprises can implement load balancing in several different ways and tailor it to benefit or emphasize specific traffic goals:

Hardware. This is a traditional physical box of circuitry connected within the physical network. A hardware load balancer can include chipsets that are designed specifically to handle traffic at full network line speeds

Software. Software installed onto a regular enterprise server also can perform load balancing. This is typically far less expensive than dedicated hardware load balancers, and upgrades are typically easier than with dedicated load-balancing devices.

Virtual instances. An enterprise can package load-balancing software into a VM or virtual appliance and then deploy it on a virtualized server. This process is simpler because the load-balancing software is already installed and configured in the VM, and it can be migrated easily between virtual servers just as any other VM.

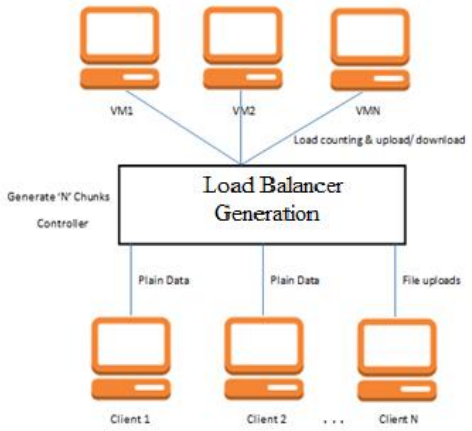

Fig. Schematic Diagram

V. CONCLUSION

The proposed work focus how multi authentication system can work in multi cloud environment. This is the first model in cloud framework which is providing multiple verification entities. Basically, the system can provide more effective results when works with multi cloud environment, it can overcome some policy base encryption methods as well as foreword security approaches.

REFERENCE

[1] Sri Shakthi Institute of Engineering and Technology, Institute of Electrical and Electronics Engineers. Madras Section, India Electronics & Semiconductor Association, and Institute of Electrical and Electronics Engineers, 1-2019 International Conference on Computer Communication and Informatics: January 23-25, 2019, Coimbatore, India.

[2] M. Kushwaha, B. L. Raina, and S. N. Singh, "2-Advanced weighted round robin procedure for load balancing in cloud computing environment," in Proceedings of the Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering, Jan. 2021, pp. 215–219. doi: 10.1109/Confluence51648.2021. 9377049.

[3] M. Bhowmik, A. Raghunandan, and B. Rudra, "3-Distributed Adaptive Video Streaming using Inter-Server Data Distribution and Agent-based Adaptive Load Balancing," in Proceedings - 16th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2020, May 2020, pp. 273–280. doi: 10.1109/ DCOSS49796.2020.00051.

[4] X. Wei and L. Wu, "4-A New Proposed Sensor Cloud Architecture Based on Fog Computing for Internet of Things," in 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Jul. 2019, pp. 615–620. doi: 10.1109/iThings/GreenCom/CPSCom/SmartDat a.2019.00120.

[5] M. Shojafar, C. Canali, R. Lancellotti, and J. Abawajy, "5-Adaptive Computing-Plus-Communication Optimization Framework for Multimedia Processing in Cloud Systems," IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1162–1175, Oct. 2020, doi: 10.1109/ TCC. 2016.2617367.

[6] L. Huo, P. Shao, F. Ying, and L. Luo, "6-The research on task scheduling algorithm for the cloud management platform of mimic common operating environment," in Proceedings - 2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science, DCABES 2019, Nov. 2019, pp. 167–171. doi: 10.1109/DCABES48411. 2019.00049.

[7] K. S. Sahoo et al., "7-ESMLB: Efficient Switch Migration-Based Load Balancing for Multicontroller SDN in IoT," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 5852–5860, Jul. 2020, doi: 10.1109/JIOT.2019.2952527.

[8] J. Fang, R. Zhang, Y. Zhao, K. Zheng, X. Zhou, and A. Zhou, "8-A-DSP: An adaptive join algorithm for dynamic data stream on cloud system," IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 5, pp. 1861–1876, May 2021, doi: 10.1109/TKDE.2019.2947055.

[9] V. J. Sosa-Sosa, A. Barron, J. L. Gonzalez, J. Carretero, and I. Lopez-Arevalo, "9-Improving Performance and Capacity Utilization in Cloud Storage for Content Delivery and Sharing Services," IEEE Transactions on Cloud Computing, 2020, doi: 10.1109/TCC.2020. 2968444.