Load Balancing and Sharing in Cloud computing for single point of failure

Vanipenta Pavan Kumar Reddy
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21215@bl.students.amrita.edu

Rejeti Kartik
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21170@bl.students.amrita.edu

Peta Sandeep
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21156@bl.students.amrita.edu

Revanth Krishna Verma
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21155@bl.students.amrita.edu

***Abstract*—Cloud services may face server overloading and they can cause delay of fractions of seconds that can cause excessive memory utilization and wastage of time. Hence the need for load balancing algorithms in the distributed network. This paper aims to find better dynamic load balancing techniques to address single point of failure and the sharing of resources of the memory. This involves the use of CloudSim simulator with Prometheus for monitoring and Grafana to display the results. The study shows that the Least Connected Network has shown the best memory and CPU utilization and great response time.**

***Index Terms*—Cloud, Load Balancing, Dynamic, Single point of failure, CloudSim, Least Connected Network**

I. Introduction

The cloud service has been extremely useful for performing tasks remotely and safely with a relative common memory and the ability to access data with the help of the internet. However during high load, i.e. when the server is receiving more requests than it can handle, it can cause massive delays in the system and could lead to crashes. Considering the issues that are, there is a need for load balancing algorithms to deal with it. Load balancing considers two different kinds of target policies, either the overloaded nodes gives the excess processes to underloaded nodes, or underloaded node requests for processes that are in waiting state.

All the nodes have a queue length which may be static or dynamic that takes processes and others outside of the queue will remain in the waiting state. The nodes however cannot deal with the single point of failure situation. This situation occurs where the program or process may be dependent on one node or one sub-task and the resource that is being used up is up to its limit. This is one of the biggest disadvantages of transfer data across cloud services.

Hence, the study proposes a comparison of multiple load balancing algorithms including Equal Spread Algorithm, Least Common Network algorithm and Weighted Round Robin algorithm to check for the way it balancing the processes. It is simulated in CloudSim with Prometheus to track the target and Grafana to show the results with graphs with metrics to prove the effectiveness of the algorithm. This ensures the safety and accuracy of the load balancing and not losing the data. The Data brokers are hence used with Java Virtual Machines along with cloudlets to execute the algorithms.

II. Literature Survey

Priya et al. [1] introduces a fuzzy-based multidimensional resource scheduling and queuing network for enhancing the load balancing and resource scheduling in cloud computing. Mainly uses the fuzzy logic to enhance the multidimensional resource allocation and decreases the response time. Nowadays techniques like Scalable Task Management and Scalable Workload Driven can have their problems solved. The proposed model is executed in Cloudsim, results are calculated based on the comparison made with older methods. Proposed method performed a 7% gain in resource scheduling efficiency and 35.5% faster response time.In future more privacy-preserving techniques are to be added for data exchange.

Deepa et al. [2] presents a new algorithm for load balancing in cloud computing mainly focuses on increasing the resource distribution efficiently across different servers. The proposed algorithm uses web engineering concepts to prioritise the client requests based on the data centre priorities. For comparison many older algorithms were used. The performance of the new method shows significantly better results in response time and utilisation of resources than other algorithms. The new method also make sure mo resources are left behind. In future the algorithm need to be more enhance on privacy features.

Mohit et al. [3] introduces a dynamic load balancing algorithm for optimising the cloud computing tasks by distributed the resources among the Virtual Machines. The novel algorithm targets on enhancing the resource utlization and reduces makespan time and also make more efficient solution for changing the cloud computing more rapidly. Benchmark

algorithms like Shortest Job First, Min-Min are used to compare it with novel approach. Metrics such as Average Resource Utilisation Ratio is used. The novel method performed better than benchmark algorithms. In future QoS parameters can be incorparated and can be exposed to various cloud infrastructures.

Shahbaz Afzal et al. [4] discusses on the issues faced in load balancing in cloud computing and mainly deals with on increasing the efficiency on handling the different workload across different Virtual machines and also the reducing the resource wastage. The proposed method is comparing with older load balancing techniques. Results tells that proactive dynamic approaches are fully dynamic with task scheduling being the most common feature. The main comparison is in the Distribution of Scheduling Traits, Algorithm Complexity, State of Algorithms and Objective Approaches. The main research gap is in the algorithm complexity that 80% of studies did not take it into consider.

Himanshu Rai et al. [5] explore the importance and the load management in cloud computing, examining the several load balancing algorithms that have been presented by researchers and highlighting the critical factors are that must be taken into account when putting such algorithms into practice in cloud systems. The issue at hand is related to the difficulties of cloud computing systems encounter while handling a high amount of the user requests, which makes it effective load control necessary to preserve peak performance. A weakness in the load balancing solutions available today necessitates the development of a novel algorithm capable of distributing the workload among the cloud infrastructure's servers in an equitable manner. The methodology proposed involves a novel load balancing algorithm that prioritizes virtual machine (VM) distribution and task allocation is developed. This algorithm was backed by a mathematical model that determines VM capacity and loads virtual machines accordingly. Simulation outputs produced with tools such as Cloud Sim illustrate the efficiency in the algorithm in this study in terms of bettering resource use, decreasing reaction times, and increasing overall throughput. In the future the work will need to be done on improving the development of trust between users and cloud providers, improving the load balancing algorithm to handle new challenges in cloud computing, and broadening its scope to include scalability, flexibility, and energy efficiency in cloud environments.

Tahira Islam et al. [6] investigate for the effectiveness of load balancing algorithms in the cloud computing. In order to examine task execution times under space-shared and time-shared scheduling rules, the study analyses three core algorithms: First Come First Served (FCFS),and Shortest Job First (SJF), and Least Connection (LC). The popular open-source framework Cloud Sim makes it possible to simulate cloud infrastructures and services. The study highlights how crucial load balancing is for enhancing system performance since then it prevents overloading by distributing work for loads among Virtual Machines (VMs) in an equitable manner. Findings show that under lower loads, LC performs better than SJF and FCFS, underscoring the need of efficient load balancing in Cloud system .The study also underscores the challenges and for benefits of Cloud computing, emphasizing the need for scalable and flexible load balancing policies to the enhance system stability, performance, and fault tolerance.

Einollah Jafarnejad Ghomi et al. [7] discuss on the many methods that have been applied for resolving load balancing on the cloud. It reviews 10 studies on load balancing and trying to address various challenges such as virtual machine migrations, single point of failure, spatial distribution of cloud nodes and energy management among others. The key insights that has been found is that the use of Hadoop MapReduce, application of agent based algorithms or creation of application specific techniques does show overall improvement of the system to address failures. These system have been tested either in real time or on simulators with metrics such as makespan, throughput, and energy ultilization among others. It also confirms that a lot of programs still struggle to address the single point of failure.

Sambit Kumar Mishra et al. [8] talk about the cloud architecture for the load balancing algorithms and the overall criteria each set of systems use. The study compares multiple algorithms including Genetic Algorithm, Minimum Execution Time algorithm, Simulated Annealing, Tabu Search, Minimum Compilation Time algorithm and Switching algorithm and the use of metrics such as fault tolerance, scalabiltiy, associated cost and many more. Among the traditional methods, Minimum Compilation Time algorithm performs the best among all using Cloud Sim simulator. These algorithms do face issues regarding the single point and virtual machine shutdown errors.

U.K. Jena et al. [9] propose an algorithm for load balancing in cloud systems. The study proposes a meta heuristic algorithm which is a hybridized learning with modified particle swarm optimization (MPSO) and Q-learning to add reinforcement rewards into the system. The proposed model has been compared to MPSO and Q-learning on the basis of Makespan, throughput, standard deviation, and energy utilization. The proposed method does better than the other models. The issue that the proposed system has is the problem of dynamic tasks which have dependencies.

Pawan Kumar et al. [10] talk about the issues of load balancing techniques and which is present among various studies. It mentions about the need for load balancing, types, its effectiveness and its measurement. It reviews static load balancing techniques and it shows that the homogeneous execution, scalabilty and response times are the biggest issues facing it. It shows that the major issues facing dynamic load balancing are the high migration time, power consumption, single point of failure and the transmission rate. The simulators also have own weaknesses, ranging from not being a complete network, to packet level communication unsupported and scalability.

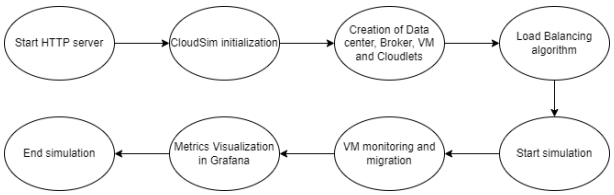


Fig. 1: Caption

III. Methodology

A. Cloud Computing and Cloudlets

The load balancing in the dynamic state in single point of failures is needed for the smooth working of cloud service. The study works on load balancing algorithms and how to deal with the single point of failure while in simulation. There are some key terminologies that has to be explained. The data center is the physical location where the data and computation machines are stored. It is where all the main servers are present. The Virtual Machine is a computer which uses the software to execute the programs but does not have its own hardware and it is sharing with some other physical system. In terms of cloud computing and distributed systems, it is able to access memory without causing issues to the other system as it mimics the working of an actual system. Cloudlets are small scale data centers which can created from a virtual machine and one host could have many cloudlets. Theses cloudlets are great for executing smaller tasks and can mimic the load overloading very easily.

B. Least Connected Network

The Least Connected Network Algorithm is a distributed load balancing algorithm where the server with the least amount of active cloudlets will be given the excess waiting tasks. In this scenario, the Virtual Machine (VM) with the least running cloudlets i.e. a receiver based target system, gets the task. The counter of each cloudlet is incremented when assigned and completed.

C. Weighted Round Robin

The weighted Round Robin algorithm is an extension to the Round Robin algorithm for spreading the incoming requests from client across the server according to the capacity of the VMs It distributes the cloudlets based on the relative weight that has been assigned by administrators. This gives an order of the cloudlets in a relative rotating order.

D. Equal Spread Algorithm

The Equal Spread Algorithm has been created to guarantee that every server gets an equal share of the tasks to be executed. In this case, there is always monitoring of the load of each VM and the cloudlets are mapped to the VMs in such a manner that the load is meant to be distributed equally in such a manner that can be switched constantly.

E. Experimentation

The experiment has been applied on the CloudSim simulator which is used as the major execution stage with Java language as the base. The initialization of the cloud simulator, allows the creation of datacenter along with the hosts, VMs and cloudlets. The number of hosts is 10, the number of VMs is set to 10 and number of cloudlets is also set to 10. The respective cloud has been in-built with many libraries to run the simulation with no issue. The respective load balancing method has been put and has been employed at different target localhosts. The simulation is then run and the different VMs allocate the cloudlets automatically and the target host shows the details of the intricacies of the function. Fig. 1 shows the entire workflow of the system.

F. Monitoring and Tracking

The tracking of pace of each algorithm is done using Prometheus, a software which is helpful in timing the network and its functionality. The target and input is interactively shown and this allows the creation of an entire prometheus workspace which is useful for the separation of each algorithm. Finally, the workspace is given to Grafana which visualizes the results obtained and shows the metrics of each algorithm in real time. For mitigation of VMs running out of service, utilization threshold is kept at 80%, selects the VMs that are not over-provisioned which gives the overview of the system and how it can be solved.

IV. Results

The algorithms have been compared based on the following metrics: HTTP response time, Memory Utilization, and CPU utilization and overall which model takes the least resource for load balancing. The HTTP response times for all three algorithms are relatively the same which means that it connects to the internet at the same time and there is no major issue in it.

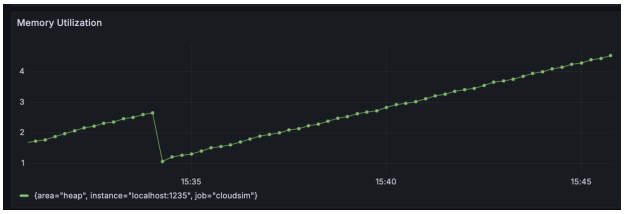
For the weighted round robin algorithm, the memory utilization increases over time and only stops increasing when the load has been balanced. The CPU utilization shows that for most of its time, some VMs become idle for a while and never fully recovers to full utilization and hence considered a way more time consuming algorithm. Fig. 2 shows the results of WRR approach.

For the least connected network algorithm, the memory utilization increases and decreases but it much more gradual decrease indicating that the load balancing is doing its job. The CPU utilization tells different story where the value is low from the start and it increases when the load is overwhelming one VM. This proves the effectiveness of the algorithm to share resources extremely. Fig. 3 shows the results of the LCN method.

For the equal spread algorithm, the memory utilization increases and decreases but it much more rapid decrease indicating that the load balancing is working. The CPU utilization shows the utilization to change slightly more than the least connected algorithm. So when the HTTP response

(a) Caption for the first image

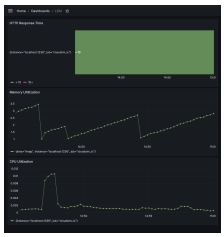
(b) Caption for the second image



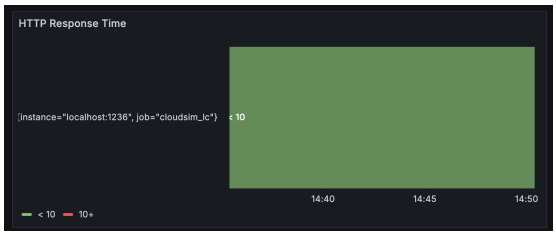
(c) Caption for the third image

(d) Caption for the fourth image

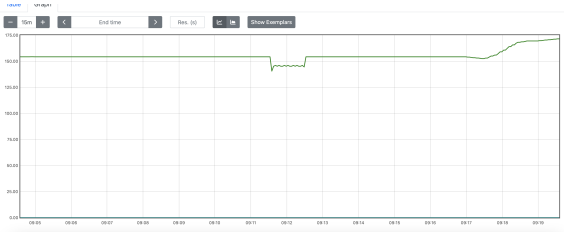
Fig. 2: Overall caption for all subfigures.



(a) Caption for the first image

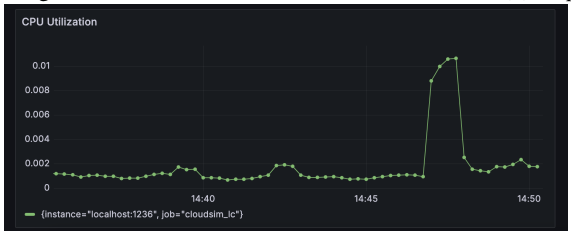


(b) Caption for the second image



(c) Caption for the third image

(d) Caption for the fourth image



(e) Caption for the fifth image

Fig. 3: Overall caption for all subfigures.

is checked, it is shown that the time is erratic which shows the inconsistency of the algorithm to connect to the cloud. To check the voracity of this claim, it is checked with and without single point of failure. It shows that the addition of mitigating the single point of failure does help as the response time is consistent and the memory utilization is good. However, the simple fact is the CPU utilization provides the highest spikes which means that the model still struggles on this aspect.

V. Conclusion

The algorithms for load balancing of the show clear and distinguished results. In terms of CPU utilization, the equal spread algorithm and the least connected network is very low and hence does not overburden the CPU. However, the weighted round robin has a high amount of resource utilization which has a higher chance of overloading. In terms of memory utilization, all models exhibit the highest memory usage, weighted robin algorithm still has the highest memory utilization, which indicates the higher workloads on the system.

Overall the equal spread algorithm keeps the CPU usage low, but it is always inconsistent in memory usage. The least connected network is suitable for keeping the utilization low and balancing accurate. The weighted round robin manages to maintain a low response time but uses a lot of memory and CPU and the balancing is not as strong compared to other methods. Overall the least connected network provides with the best load balancing scenario with consistent memory and CPU utilization keeping response times low.

Future work for the project could include extending the complex load balancing rules with supplementing load balancing functionality including machine learning forecasting techniques and predictive analytics and applying artificial intelligence to load balancing. This can go together with live parameterization capabilities that could be attached to widely utilized public cloud platforms such as AWS, Azure, Google Cloud to offer users insights on the operation of the system under various loads. Other changes that may require further consideration and knowledge and a more inclusive view of the costs and effective means by which people can gain the best from the resources available and lower operating expenses. Finally, the system in terms of security issues and fault tolerance may define the framework as more valuable and secure depending on a situation with clouds for resource management.

References

[1] V. Priya, C. S. Kumar, and R. Kannan, "Resource scheduling algorithm with load balancing for cloud service provisioning," *Applied Soft Computing*, vol. 76, pp. 416–424, 2019.

[2] D. Bura, M. Singh, and P. Nandal, "Analysis and development of load balancing algorithms in cloud computing," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 13, no. 3, pp. 35–53, 2018.

[3] M. Kumar and S. C. Sharma, "Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing," *Procedia computer science*, vol. 115, pp. 322–329, 2017.

[4] S. Afzal and G. Kavitha, "Load balancing in cloud computing–a hierarchical taxonomical classification," *Journal of Cloud Computing*, vol. 8, no. 1, p. 22, 2019.

[5] H. Rai, S. K. Ojha, and A. Nazarov, "Cloud load balancing algorithm," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 861–865, 2020.

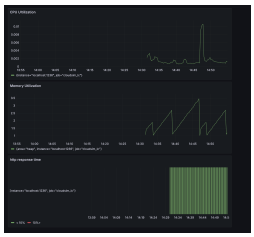
[6] T. Islam and M. S. Hasan, "A performance comparison of load balancing algorithms for cloud computing," in *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*, pp. 130–135, 2017.

[7] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 50–71, 2017.

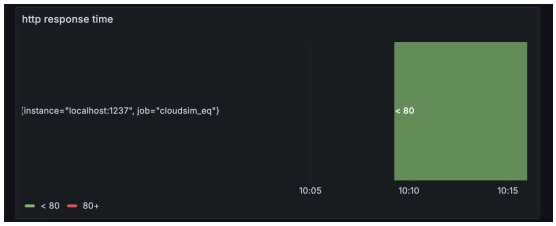
[8] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: a big picture," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp. 149–158, 2020.

[9] U. K. Jena, P. Das, and M. R. Kabat, "Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2332–2342, 2022.

[10] P. Kumar and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: A survey," *ACM computing surveys (CSUR)*, vol. 51, no. 6, pp. 1–35, 2019.



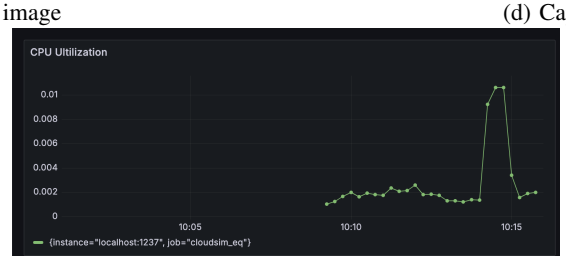
(a) Caption for the first image



(b) Caption for the second image

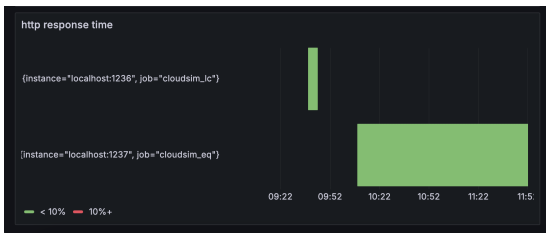
(c) Caption for the third image

(d) Caption for the fourth image

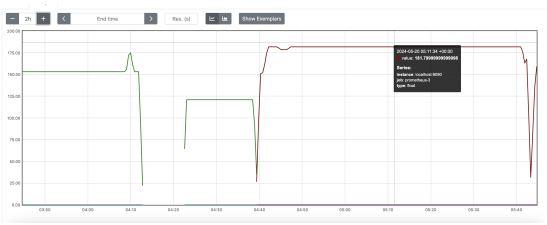


(e) Caption for the fifth image

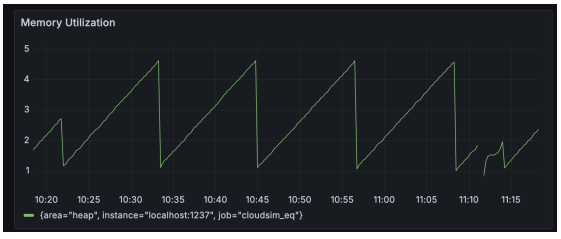
Fig. 4: Overall caption for all subfigures.



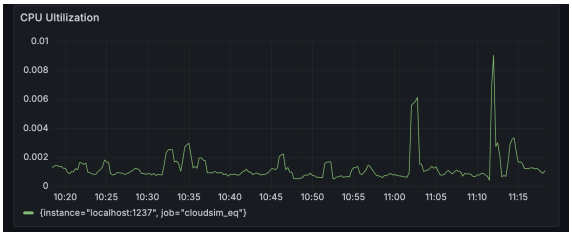
(a) Caption for the first image



(b) Caption for the second image



(c) Caption for the third image



(d) Caption for the fourth image

Fig. 5: Overall caption for all subfigures.