Journal of Nano- and Electronic Physics
Vol. **15** No 3, 03023(4pp) (2023)

Журнал нано- та електронної фізики
Том **15** № 3, 03023(4cc) (2023)

Load Balancing Strategies for Cloud Computing: A Simulation-Based Study

Navneet Kumar Rajpoot*, Prabhdeep Singh†, Bhaskar Pant‡

Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, India

(Received 25 April 2023; revised manuscript received 17 June 2023; published online 30 June 2023)

The CloudAnalyst simulation tool offers various load balancing techniques that can be utilized for efficiently distributing tasks. This research paper explores various load balancing techniques by utilizing the CloudAnalyst simulation tool. The research paper intends to evaluate and contrast the effectiveness of conventional load balancing techniques. While gathering information, the technique entails the simulation of diverse cases with varying parameters, including server capacity, workload, and network latency. The results are compared and contrasted to demonstrate that each technique for load balancing has strengths and weaknesses, and that the most suitable technique should be chosen based on the specific scenario. The Round Robin algorithm is easy to implement but may not be suitable for all scenarios. The Least Connection algorithm is suitable for scenarios where server capacity is not uniform. The IP Hash algorithm is useful for stateful applications, while the Weighted Round Robin algorithm is suitable for scenarios where servers have different capacities. The Least time algorithm is useful for scenarios where processing time is critical. The proposed framework for this research paper is based on nature-inspired load balancing using the CloudAnalyst simulation tool. The framework aims to develop a load balancing algorithm that can adapt to changing workload patterns in real-time.

Keywords: Ant Colony Optimization, Cloud computing, CloudAnalyst, Load balancing, Nature-inspired algorithm.

DOI: 10.21272/jnep.15(3).03023 PACS number: 07.05.Tp

1. INTRODUCTION

Load balancing ensures that the workload is distributed evenly across multiple servers, thus avoiding overloading and enhancing the entire system's efficiency. Load balancing is especially important in cloud computing, where multiple users access the same resources simultaneously [1]. CloudAnalyst simulation tool is a cloud simulation software designed to optimize and improve cloud computing performance [2]. This tool effectively simulates various cloud scenarios, including load balancing. This tool has the ability to simulate various parameters that can impact cloud computing performance, including workload, server capacity, and network conditions [3]. This paper aims to explore load balancing by utilizing the CloudAnalyst simulation tool and its diverse range of parameters.

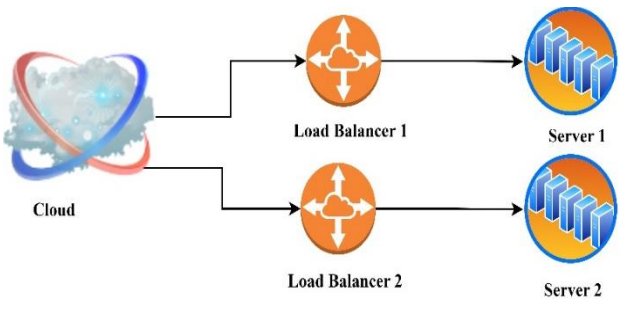

Fig. 1 – Load balancing in the cloud

CloudAnalyst facilitates load balancing simulations to evaluate different scenarios and identify the most effective load balancing approach. The process entails creating simulations to analyze the workload and server capacity to identify the most effective utilization of resources [4]. This simulation tool employs advanced techniques to ensure an equitable workload distribution among several servers. CloudAnalyst offers several choices for performing load balancing simulations, which can be explored based on individual preferences and requirements.

In this simulation, virtual machines are distributed among actual servers according to the servers' workloads and available resources. This simulation is useful for deciding where to put virtual machines to balance loads evenly [5].

This simulation simulates the server's capacity and workload to identify the most effective resource allocation. The simulation takes into account a range of parameters [6].

For running a load balancing simulation using CloudAnalyst, some different parameters are required.

The workload can be defined as the volume of data or requests that a system receives. CloudAnalyst offers a range of solutions for simulating workloads using different parameters.

Server capacity is the maximum amount of workload that a server can manage. CloudAnalyst offers the ability to simulate server capacity using various parameters, including CPU utilization, memory utilization, and network traffic.

The term "algorithm" is related to the load balancing technique utilized for distributing the workload. CloudAnalyst offers the ability to simulate multiple techniques to identify the most effective approach for

* navneetrajpootgeu@gmail.com
† ssingh.prabhdeep@gmail.com
‡ bhaskar.pant@geu.ac.in

load balancing [7].

Network conditions are related to the network's performance that connects the servers. CloudAnalyst offers the ability to simulate network conditions using parameters such as latency, packet loss, and bandwidth.

Load balancing simulation using CloudAnalyst provides various benefits, such as: load balancing simulation helps in determining the best allocation of resources to achieve optimal performance [8]. Load balancing simulation helps in reducing the cost of resources by allocating resources efficiently [9]. The system's overall performance is enhanced by load balancing simulations, which equally distribute the workload throughout several nodes. [10].

2. COMPARISON OF THE LOAD BALANCING ALGORITHMS USED IN CLOUDANALYST

CloudAnalyst provides several load balancing algorithms that can be used to distribute workloads across multiple servers. This paper will compare the load balancing algorithms used in CloudAnalyst and their configuration.

Round Robin algorithm distributes workload across multiple servers in a cyclic order. This algorithm allocates the workload to the next available server in the sequence [11]. This algorithm is easy to implement and does not require any complex configurations. However, it may not be suitable for all scenarios as it does not consider server capacity or workload [12].

The tasks are distributed between servers so that the one with the fewest ongoing sessions receives the most of it. Using this approach, the load balancer determines which system has the fewest ongoing sessions as well as sends incoming sessions to that system [13]. This algorithm is suitable for scenarios where server capacity is not uniform. It requires configuring the maximum number of connections a server can handle.

The IP Hash algorithm utilizes the source IP address for distributing the workload. This algorithm utilizes the load balancer to determine the hash of the source IP address and then assigns the workload to the server that corresponds with the hash value. This algorithm can direct the same source IP address to a consistent server, which can be advantageous for stateful applications [14]. There are certain parameters that need to be configured, such as the hash function and the number of servers.

Weighted Round Robin Load Balancing Algorithm

The Weighted Round Robin algorithm distributes workload among multiple servers, considering their respective weightage. This algorithm entails assigning a weight to each server, which helps to determine the workload proportion assigned to that server [15]. This algorithm can be particularly helpful when servers have varying capacities. To optimize performance, it may be necessary to adjust the server weightage and limit the connections each server can handle.

Least Time Load Balancing Algorithm

The Least Time algorithm is designed to allocate workload to the server with the shortest processing time. In this algorithm, the load balancer distributes requests to each server as well as measures the time taken to process them. According to the system's algorithm, the request is assigned to the server with the most efficient processing time [16]. This algorithm can be particularly advantageous when time is of its essence. To ensure optimal server performance, it is recommended to configure the maximum number of connections along with request processing timeout [17].

3. RELATED WORK

Over the years, researchers have proposed various load balancing algorithms, each with advantages and disadvantages [18]. Several studies have investigated load balancing algorithms and their performance in cloud computing. Utilizing the cloudSim modeling framework, B.L. Raina et al. Analyzed several load balancing strategies. In terms of response time as well as performance, they discovered that the weighted round robin algorithm worked best [19]. Similarly, U. Tyagi et al. evaluated various load balancing methods, which were simulated with the help of the ns-3 tool for modeling as well as simulations. They concluded that the weighted round robin algorithm offered the most favorable results regarding response time and efficiency [20]. Other studies have focused on improving load balancing algorithms using machine learning techniques. T.M. Bhalodia et al. proposed a load balancing technique based on a neural network that adapts to changing workload patterns.

4. METHODOLOGY

The methodology used in this research paper involves simulating load balancing scenarios using the CloudAnalyst simulation tool. The simulation tool includes a load balancer, servers, and workloads that can be configured to simulate various scenarios with different parameters. The simulations were conducted by varying parameters such as server capacity, workload, and network latency. The workload was generated using a custom script that emulated real-world traffic patterns. The simulation results were collected and analyzed using statistical software. The performance of each load balancing algorithm was evaluated. Multiple simulations were conducted for each scenario to ensure the validity of the results, and the results were compared to ensure consistency. Compassion analysis was also conducted to evaluate the impact of different parameters on load balancing performance.

5. THE PROPOSED FRAMEWORK

The proposed framework for this research paper is based on nature-inspired load balancing using the CloudAnalyst simulation tool. The framework aims to develop a load balancing algorithm that can adapt to changing workload patterns in real-time using techniques inspired by natural systems. The ACO algorithm simulates the pheromone trails that ants leave when foraging for food. The pheromone trail is a signal that other ants can follow to locate the food source. In the process of load balancing, the pheromone trail can represent the workload on each server. The proposed algorithm uses the ACO algorithm to balance the workload by directing incoming requests to the server with the lowest work-

load. The proposed framework has tested through simulations utilizing the CloudAnalyst simulation tool. We have evaluated the suggested algorithm's efficacy concerning other well-known load balancing strategies. Metrics like response time, throughput, and error rate have applied to the simulation findings.

6. EXPERIMENTAL SETUP USING CLOUDANALYST

The experimental setup for this research paper uses the CloudAnalyst simulation tool to evaluate the performance of different load balancing algorithms. The simulation tool includes a load balancer, servers, and workloads that can be configured to simulate various scenarios with different parameters. To conduct the experiments, we first set up the simulation environment by configuring the number of servers, server capacity, and network latency. Then we selected the load balancing algorithms to be evaluated. We also generated workloads using a custom script that emulated real-world traffic patterns. We conducted multiple simulations for each scenario to ensure the consistency and reliability of the results. The simulation results were collected and analyzed using the CloudAnalyst simulation tool. The load balancing algorithms have been evaluated using various metrics, including response time, throughput, and error rate. To enhance the accuracy of the outcomes, we carried out a comparative analysis to evaluate the impact of various parameters on load balancing performance. We also compared the results of the different load balancing algorithms to identify their strengths and weaknesses and their suitability for different scenarios.

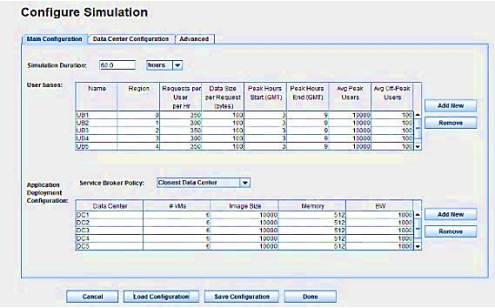

Fig. 2 – Data center configuration in Cloud analyst

Fig. 3 – Simulation Executon in Cloud Analyst

Table 1 – Response time by region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UserBase1	50.04	41.11	62.36
UserBase 2	50.28	40.14	61.89
UserBase 3	50.03	40.38	61.63
UserBase 4	50.12	39.13	61.88
UserBase 5	50.17	38.39	61.17

7. CONCLUSION

Distributing workloads across different servers using load balancing helps to maximize cloud performance, expandability, and availability. This study uses the CloudAnalyst simulation tool to compare and contrast several load-balancing techniques. The weighted round robin algorithm delivered better response time and throughput performance compared to the least time approach, which performed the lowest in our testing. Furthermore, we devised a load-balancing system that takes cues from nature and is based on ant colony optimization to adjust dynamically to fluctuating workloads. Results from the simulations demonstrated that the suggested approach provided better response time and throughput than the state-of-the-art load balancing strategies.

REFERENCES

1. S.A. Narale, P.K. Butey, *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (2018).
2. K.D. Patel, T.M. Bhalodia, *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (2019).
3. M. Kushwaha, B.L. Raina, S. Narayan Singh, *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (2021).
4. Y. Bo, *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)* (2022).
5. U. Tyagi, V. Bansal, S. Singhal, T. Gupta, *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)* (2022).
6. S.M. Irfan, H. Rathore, H. Bisen, D. Kumar, S. Kumar, K. Sharma, 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) (2022).
7. A. Jyoti, M. Shrimali, R. Mishra, *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (2019).
8. S. Paul, M. Adhikari, *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2018).
9. H. Rai, S.K. Ojha, A. Nazarov, *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (2020).
10. S. Shukla, A.K. Singh, V. Kumar Sharma, *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (2021).
11. M. Zedan, G. Attiya, N. El-Fishawy, *2021 International Conference on Electronic Engineering (ICEEM)* (2021).
12. A. Srivastava6 N. Kumar, *Int. J. Inform. Technol.* **15** No 1, 107 (2022).

13. M. Sumathi, N. Vijayaraj, S.P. Raja, M. Rajkamal, *Int. J. Inform. Technol.* **15**, 1357 (2023).
14. R.R.K. Chaudhary, K. Chatterjee, *Int. J. Inform. Technol.* **14** No 6, 3109 (2022).
15. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism, 3rd ed., vol. 2* (Oxford: Clarendon: 1892).
16. H.K. Yugank, R. Sharma, S.H. Gupta, *Int. J. Inform. Technol.* **14** No 5, 2549 (2022).
17. R.S. Pawar, D.R. Kalbande, *Int. J. Inform. Technol.* **15**, 595 (2023).
18. N. Garg, M.S. Obaidat, M. Wazid, A.K. Das, D.P. Singh, *ICC 2021 – IEEE International Conference on Communications* (2021).
19. N. Garg, M. Wazid, A.K. Das, D.P. Singh, J.J.P.C. Rodrigues, Y. Park, *IEEE Access* **8**, 95956 (2020).
20. S. Pundir, M. Wazid, D.P. Singh, A.K. Das, J.J.P.C. Rodrigues, Y. Park, *IEEE Access* **8**, 3343 (2020).
21. N. Singh, D.P. Singh, B. Pant, International *J. Math., Eng. Manag. Sci.* **4** No 5, 1239 (2019).

Стратегії балансування навантаження для хмарних обчислень: дослідження на основі моделювання

Navneet Kumar Rajpoot, Prabhdeep Singh, Bhaskar Pant

Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, India

Інструмент моделювання CloudAnalyst пропонує різні методи балансування навантаження, які можна використовувати для ефективного розподілу завдань. У цьому дослідницькому документі досліджуються різні методи балансування навантаження за допомогою інструменту моделювання CloudAnalyst. Дослідницький документ спрямований на оцінку та порівняння ефективності звичайних методів балансування навантаження. Під час збору інформації ця техніка передбачає моделювання різноманітних випадків із різними параметрами, включаючи потужність сервера, робоче навантаження та затримку мережі. Результати порівнюються та порівнюються, щоб продемонструвати, що кожен метод балансування навантаження має сильні та слабкі сторони, і що найбільш прийнятний метод слід вибирати на основі конкретного сценарію. Алгоритм Round Robin простий у реалізації, але він може не підходити для всіх сценаріїв. Алгоритм найменшого підключення підходить для сценаріїв, коли потужність сервера неоднакова. Алгоритм хешування IP корисний для додатків із збереженням стану, тоді як алгоритм Weighted Round Robin підходить для сценаріїв, де сервери мають різну потужність. Алгоритм найменшого часу корисний для сценаріїв, де час обробки є критичним. Запропонована основа для даної дослідницької роботи базується на природному балансуванні навантаження за допомогою інструменту моделювання CloudAnalyst. Інфраструктура спрямована на розробку алгоритму балансування навантаження, який може адаптуватися до змінних моделей робочого навантаження в режимі реального часу.

Ключові слова: Оптимізація, Хмарні обчислення, CloudAnalyst, Балансування навантаження, Природний алгоритм.