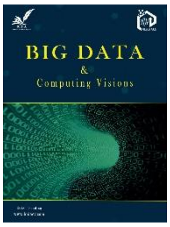
Big Data and Computing Visions

www.bidacv.com

Big. Data. Comp. Vis. Vol. 3, No. 2 (2023) 45–49.



Paper Type: Original Article

An Introductory Guide to Cloud Load Balancing Techniques

Mingyue Wang*

School of Computer and Information, Lanzhou University of Technology, Gansu Provice, China; wang.mingyue9811@gmail.com.

Citation:

Wang, M. (2023). An introductory guide to cloud load balancing techniques. *Big data and computing visions, 3*(2), 45-49.

Received: 07/01/2023 Reviewed: 09/02/2023 Revised: 19/03/2023 Accepted: 15/04/2023

Abstract

The goal of research in the intriguing area of cloud computing is to discover the most effective solution and method for sharing and protecting data. Load Balancing (LB) on public infrastructure using cloud computing's ideal geographic location. In cloudy environments, LB is widely used as a visitor control approach. Cloud requests look for resources to support performance. The resources are frequently things like bandwidth, processing power, and storage. LB is the process of effectively allocating these resources to each competing job. This study will offer a thorough analysis of cloud load-balancing methods.

Keywords: Scheduling, Load balancing, Cloud computing.

1 | Introduction

Cloud computing has become one of the most practical solutions to problems that may require extensive calculation. The cloud provides a way to share resources and services with people as needed [1]. Without being aware of location differences, one can access virtualized resources and services. The methods offered by cloud computing take into account run-time requests for computing resources like storage, availability, software, and so forth [2]. The cloud platforms make distinctions between the various service styles, rates, and Quality of Service (QoS) in addition to performance [3]. Because of this reality, cloud users have the freedom to choose their target structure from a wide variety of cloud platforms. Even so, this increases the challenge of interoperability among the unique clouds [4]. The development of effective carrier provisioning policies is the main concern in the study of clouds. The clouds of today live in an open environment where changes occur continuously, randomly, and without warning. Game theoretic approaches enable in-depth analytical understanding of the provider provisioning challenge in this environment [5]. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are the three types of facilities that cloud computing usually always provides SaaS [6].

A network built on the internet underpins cloud computing. A cloud is a group of services. On-demand services are offered through cloud. Hardware service, software service, and network service

Corresponding Author: wang.mingyue9811@gmail.com

https://doi.org/10.22105/bdcv.2023.392179.1121

are the three main services offered through the cloud [7]. Modern fields like utility computing, service-oriented architecture, the internet, and clients all center around cloud computing. Cloud computing is still in its early stages of development and faces many problems and difficulties. One of the most crucial factors in determining your present successful execution is cloud scheduling [8]. The term "scheduling" refers to a group of guidelines that regulate the sequence in which work is likely to be completed by a computer system. Job scheduling is one of the various parties involved with the scheduling algorithms that are present in distributed computing systems [9]. The primary benefit of a job scheduling algorithm is that it allows for the quickest process throughput and good high-performance computing. A proper scheduling policy allows for maximum resource usage. Scheduling controls CPU memory availability [10].

1.1 | Load Balancing in Cloud Computing

The Load Balancing (LB) is the process of distributing the load among various resources in any system. Therefore, load need to be distributed over the resources in cloud-based architecture so that each resource does approximately the equal amount of task at any point of time [11]. The basic need is to provide some techniques to balance requests to provide the solution of the application faster. All cloud vendors are based on automatic LB services, it allows clients to increase the number of CPUs or memories for their resources to scale with increased demands [12]. These services are optional and depend on the client's business needs. So, the LB serves two important needs, firstly to promote availability of cloud resources and secondarily to promote performance [13]. It is crucial to understand a few of the main objectives of LB algorithms in order to balance the resources:

I. Cost effectiveness: the initial goal is to increase system performance generally at a fair price [14].
II. Flexibility and scalability: the size or topology of the distributed system in which the method is applied may change. Thus, the algorithm must be scalable and adaptable enough to manage such changes without difficulty [15].
III. Priority: despite providing equal service for all jobs, regardless of where they came from, scheduling of the resources or jobs must be done beforehand through the algorithm itself for better service to the significant or highly prioritized jobs [16].

2 | Literature Review

The works in this section were previously proposed by a number of researchers. Here, several typical strategies that are effective in terms of response time, data centre processing time, and cost are also mentioned. An innovative LB technique called VectorDot was proposed in the work of [17]. In a flexible data centre with built-in server and storage virtualization technologies, this algorithm manages the hierarchical complexity of the datacenter and multidimensionality of resource loads across servers, network switches, and storage. The research conducted Panda et al. [18] proposed a cloud control technique called CARTON that combines the use of LB and Distributed Rate Limiting (DRL). The DRL is used to ensure that the resources are divided in a way to maintain a fair resource allocation [19]. The LB is used to evenly distribute the workloads to multiple servers in order to minimize the related expenses. Tareen et al. [20] used adaptive live migration of virtual machines to solve the intra-cloud LB issue among physical hosts. By using shared storage to distribute the load among servers in accordance with their CPU or IO consumption, the LB model is created and put into use to shorten the time required to migrate virtual machines. An event-driven LB system for real-time massively multiplayer online games was presented in work by [21].

After receiving capacity events as input, the algorithm additionally analyses its constituent parts in light of the resources and the overall condition of the game session before producing the LB actions for the game session. Mohapatra [22] suggested a load-balancing scheduling technique for virtual machine resources based on historical data and the present system status. By utilizing a genetic algorithm, the suggested technique achieves the optimal load balance and minimizes dynamic migration. In a distributed virtual machine/cloud computing environment, Mohapatra [23] suggested a Central LB Policy for Virtual Machines (CLBVM) that distributes the load equitably. A large scale net data storage model and a SaaS model based on cloud storage are provided by the LB Virtual Storage Strategy (LBVS) that was described by [10]. A three-layered architecture is used to provide storage virtualization, and two LB modules are used to implement LB. It aids in increasing effectiveness. To accommodate users' changing needs and achieve high resource utilization, Mishra and Majhi [11] discussed a two-level job scheduling approach based on LB. The LB algorithm works by first mapping jobs to virtual machines, then virtual machines to host resources, optimizing task response time, resource usage, and overall cloud computing environment performance. Jyoti et al. [12] looked at a self-organizing LB method based on honey bees that is decentralized and nature inspired. Using local server operations, the algorithm achieves global LB. More system variety improves the system's performance, but larger systems do not boost throughput. This works well in situations where a wide variety of service kinds are needed.

3 | The Algorithm

Here is the updated proposed throttled algorithm:

Input: r1, r2, rn data centre requests.

Virtual machines: vm1, vm2, and vmn are available.

Output: the available virtual machines vm1, vm2, ..., vmn are allocated to the data center requests r1, r2, ..., rn.

Step 1. Confirm that every vms allocation status is available in the vm state list. On the basis of processing time, response time, and available memory.

Step 2. Maintain a hash map table of vm. HashMap should be started with zero entries.

Step 3. A new request is received by the datacenter controller.

Step 4. The datacenter controller askes the new load balancer for the upcoming allocation. The hash map is sorted according to response time, in descending order.

Step 5. Allocate the vm if the HashMap list size is larger than the vm state list size. If not, wait for the vm to free up.

Step 6. The datacenter controller notices the load balancer of the vm deallocation when the vm has finished processing the request and has received the cloudlet response. When the virtual machine completes the request (*Step 7*).

Step 7. The data centre controller sends a notification to updated throttled that the vm id has finished the request. The hash map table is modified in accordance with updated throttled.

The algorithm comprises several steps, as outlined. First, the algorithm confirms that every virtual machine's allocation status is available in the virtual machine state list. Next, it maintains a hash map table of virtual machines, starting with zero entries. When a new request is received by the data center controller, the algorithm asks the new load balancer for the upcoming allocation. The hash map is then sorted according to response time, in descending order.

When the hash map list size is larger than the virtual machine state list size, the algorithm allocates the virtual machine. If not, it waits for the virtual machine to become available. Once the virtual machine has finished processing the request and has received the cloudlet response, the data center controller notifies the load balancer of the virtual machine deallocation. Finally, the hash map table is modified in accordance with the updated throttled algorithm.

The algorithm aims to achieve efficient resource usage by balancing the load across available virtual machines. By prioritizing response time and available memory, the algorithm ensures that data center requests are processed as quickly and efficiently as possible. However, it is important to note that the effectiveness of the algorithm may depend on specific system characteristics and workload patterns, and further research could be conducted to optimize the algorithm for specific use cases.

4 | Conclusion

By dividing the public cloud into a few sub-clouds, we define LB in this type of paper. This segmentation of the general cloud cover into a number of smaller clouds is carried out in a good geographic area. This work profitably developed the idea of cloud LB. It also offered a thorough analysis of the most widely used and current methods of cloud LB. The authors of this publication hope that the research they conducted will be useful to other researchers in the future.

References

[1] Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: a review. *Journal of King Saud university-computer and information sciences*, *34*(7), 3910-3933.

[2] Kumar, P., & Kumar, R. (2019). Issues and challenges of load balancing techniques in cloud computing: a survey. *ACM computing surveys (CSUR)*, *51*(6), 1-35. DOI:10.1145/3281010

[3] Afzal, S., & Kavitha, G. (2019). Load balancing in cloud computing-a hierarchical taxonomical classification. *Journal of cloud computing*, *8*(1), 22. https://doi.org/10.1186/s13677-019-0146-7

[4] Pradhan, A., & Bisoy, S. K. (2022). A novel load balancing technique for cloud computing platform based on PSO. *Journal of King Saud university-computer and information sciences*, *34*(7), 3988–3995.

[5] Yu, H. (2022). Difference between domestic and hostile applications of wireless sensor networks. *Big data and computing visions*, *2*(4), 149–153.

[6] Alkhatib, A. A., Alsabbagh, A., Maraqa, R., & Alzubi, S. (2021). Load balancing techniques in cloud computing: extensive review. *Advances in science, technology and engineering systems journal*, *6*(2), 860–870. DOI:10.25046/aj060299

[7] Milan, S. T., Rajabion, L., Ranjbar, H., & Navimipour, N. J. (2019). Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments. *Computers & operations research*, *110*, 159–187. DOI:https://doi.org/10.1016/j.cor.2019.05.022

[8] Kaur, A., & Kaur, B. (2022). Load balancing optimization based on hybrid heuristic-metaheuristic techniques in cloud environment. *Journal of King Saud university-computer and information sciences*, *34*(3), 813–824.

[9] Junaid, M., Sohail, A., Rais, R. N. B., Ahmed, A., Khalid, O., Khan, I. A., ... & Ejaz, N. (2020). Modeling an optimized approach for load balancing in cloud. *IEEE access*, *8*, 173208–173226.

[10] Agarwal, R., Baghel, N., & Khan, M. A. (2020). Load balancing in cloud computing using mutation based particle swarm optimization. *2020 International conference on contemporary computing and applications (IC3A)* (pp. 191-195). IEEE.

[11] Mishra, K., & Majhi, S. K. (2021). A binary bird swarm optimization based load balancing algorithm for cloud computing environment. *Open computer science*, *11*(1), 146–160.

[12] Jyoti, A., Shrimali, M., Tiwari, S., & Singh, H. P. (2020). Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey. *Journal of ambient intelligence and humanized computing*, *11*, 4785–4814.

[13] El-Morsy, S. A. (2022). Comparison between domestic and hostile applications of wireless sensor networks. *Computational algorithms and numerical dimensions*, *1*(1), 30–34.

[14] Zhou, J., Lilhore, U. K., Hai, T., Simaiya, S., Jawawi, D. N. A., Alsekait, D., ... & Hamdi, M. (2023). Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing. *Journal of cloud computing*, *12*(1), 1–21.

[15] Mohapatra, H., & Rath, A. K. (2021). Fault tolerance in WSN through uniform load distribution function. *International journal of sensors wireless communications and control*, *11*(4), 385–394.

[16] Mohapatra, H., & Rath, A. K. (2020). Nub less sensor based smart water tap for preventing water loss at public stand posts. *2020 IEEE microwave theory and techniques in wireless communications (MTTW)* (Vol. 1, pp. 145–150). IEEE.

[17] Bhosale, N., Shinde, T., & Nimbalkar, S. (2023). Load balancing techniques in cloud computing. *Vidhyayana-an international multidisciplinary peer-reviewed e-journal-ISSN 2454-8596*, *8*(si7), 709–730.

[18] Panda, H., Mohapatra, H., & Rath, A. K. (2020). WSN-based water channelization: an approach of smart water. *Smart cities—opportunities and challenges: select proceedings of ICSC 2019* (pp. 157–166). Springer Singapore.

[19] Mohapatra, H., & Rath, A. K. (2020). IoT-based smart water. *IoT technologies in smart cities: from sensors to big data, security and trust*, 63–82. https://www.researchgate.net/publication/341654237_IoT-based_smart_water

[20] Tareen, F. N., Alvi, A. N., Malik, A. A., Javed, M. A., Khan, M. B., Saudagar, A. K. J. , ... & Abul Hasanat, M. H. (2023). Efficient load balancing for blockchain-based healthcare system in smart cities. *Applied sciences*, *13*(4), 2411. https://www.mdpi.com/2076-3417/13/4/2411

[21] Mohapatra, H. (2020). Offline drone instrumentalized ambulance for emergency situations. *IAES international journal of robotics and automation*, *9*(4), 251–255.

[22] Mohapatra, H. (2009). *HCR using neural network* (Doctoral Dissertation, Biju Patnaik University of Technology). https://www.academia.edu/29846341/HCR_English_using_Neural_Network

[23] Mohapatra, H. (2019). *Ground level survey on sambalpur in the perspective of smart water* (No. 1918). DOI:10.13140/RG.2.2.24106.36806