10.48047/jocaaa.2024.33.07.21

# The Cloud-Based Hybrid Resource Utilization Load Balancing Algorithm

**C.S. Prasanna [1], Dr.Mayank Singh Parihar[2]**
[1]Research Scholar, Dr. C.V. Raman University, Kota, Bilaspur, 495113, India.
[2]Department of Information Technology, Dr. C.V. Raman University, Kota, Bilaspur, 495113, India.
[1]cspras2017@gmail.com, [2]cvrumayank@gmail.com

## 1. Abstract

Due to imbalanced workload distribution, load balancing is one of the most challenging problems in cloud computing. This paper proposes an intelligent hybrid load balancing algorithm based on resource utilization represented with an exponential moving average of both CPU and memory utilization. Experimental results indicate that this load balancing algorithm is capable of optimizing resource utilization. The discrete workload of different instance types needs to be adjusted to fit the target workload, and the target workload is calculated in terms of the number of virtual cores of the donor instance and receiver instance, the consolidation ratio of the current host, and the consolidation ratio of the current cluster. A hybrid resource utilization-based load balancing solution based on fuzzy logic together with genetic evolution is proposed, and a load balancing algorithm based on resource utilization is proposed. Experiments show that the computing time of these new algorithms is limited, which indicates that these algorithms are efficient for the cloud-based environment. [1]

This paper contributes as follows: (1) A hybrid resource utilization-based load balancing algorithm based on fuzzy logic and genetic evolution is proposed. (2) Based on the load-state prediction, a cloud-based similarity match-based algorithm is proposed within 20%. In securing the environment, it would locate and terminate the live migration VM. (3) It optimizes resources, including CPU, memory, and bandwidth, according to the virtual machine of the cloud. (4) In different approaches combined with CPU-memory and memory-CPU, the performance is evaluated. (5) The experiments demonstrate that the best instance scheme resource utilization is the load file scheme, achieving an average CPU utilization of 69.67%. (6) Experimental results demonstrate that the resource utilization can be effectively improved by balancing the load. [2]

## 2. Introduction

Cloud computing is a prevalent and emerging technology that has widely been used in different organizations. Cloud computing technologies have developed progressively, but they face numerous challenging tasks, such as resource optimization, fault tolerance, and the security of cloud data. Resource utilization is significant in assigning cloud computing resources, such as workload, storage, data query, and computational resources. Cloud computing is a set of resources that can be organized as a pool with a network of shared data used in data storage and to optimize resource utilization and sharing with the aid of the concept of load balancing. Load

10.48047/jocaaa.2024.33.07.21

balancing is the method of distributing resources between network computers to optimize resource consumption and, as a consequence, develop economic efficiency. As a result, the load balancing algorithm of cloud-based hybrid resource utilization is significant, reduces downtime, and improves resource utilization efficiency. The remainder of this report is divided into the following sections. Section II is a literature review that outlines the body of research done in load balancing technology. The goal of the literature review is introduced, continuing with a section that discusses the research topic. The significance and contributions of this report are discussed in Section III. In Section IV, the methodology for the research and algorithms is discussed, and in Section V, a case study simulation and the algorithm are shown. A parameter study of the simulations as well as the performance of the algorithm is carried out. The study conclusion is indicated in Section VI. The findings and recommendations of this thesis are provided at the end. [3]

## 3. Background and Related Work

Over the last few years, cloud computing has become a key technology provided by various service providers, with load balancing being one of the primary challenges in this field. The principal objective of load balancing in the cloud computing platform is to maintain an equilibrium condition among the various tasks, thereby presenting an overall improvement in system performance. Several studies investigate the primary effects related to the cloud computing environment. A significant amount of work deals with the various approaches in the cloud. There are certain issues with the existing methodologies; the steady-state detection mechanism in the existing approach is reactive and reduces overall performance. The suggested methodology is hybrid and can preserve overall system resources. [4]

Cloud computing is an infrastructure already established for today's world, so a new methodology will be proposed that is effective and reflects a low-cost utilization mechanism. Hence, a new approach is suggested in the present context – the Hybrid and Quality of Service-based Resource Utilization Load Balancing Algorithm. This research aims to analyze and create global load balancing mechanisms in the system design. Moreover, the contribution of load balancing is to extend the life cycle of cloud computing resources to control server overloads using various methodologies. Since the early 1980s, the cloud computing community has been exploring a wide range of methodologies. Task Scheduling based on Load Balancing instead of server, time, or processor slicing suggests keeping the entire system as a whole. The 1950s introduced the first task allocation method based on the task's need and variety, with decreasing allocations in parallel. The area of cloud computing mainly deals with performance presentation by employing Task Scheduling based on Load Balancing. A few studies on the hybrid mechanisms of load balancing were considered in previous work. Currently, the performance is supposed to be improved. However, previous studies lack an efficient investigation in this sector. Additionally, the papers published in the last few years demonstrated complete solutions. [5]

## 4. Cloud Computing and Load Balancing

Over the past few years, cloud computing has emerged as a technology that eases the usage of shared computing resources where users can host their applications. Cloud computing reduces the

10.48047/jocaaa.2024.33.07.21

user's need to install the applications on their machines. It allows users to access the applications via frameworks like web browsers. Hardware resources, storage, infrastructure, and servers are some types of cloud services provided to users for hosting their applications. The cloud architecture supports on-demand resource allocation, which allows users to rent resources based on their requirements. [6]

The computation of tasks on cloud infrastructure is becoming a key area in cloud computing. The resources in cloud environments required to serve requests from users lead to load balancing of servers. Load balancing is the strategy adopted for efficient resource management. In this strategy, the server's overhead is alleviated by assigning tasks to the servers in an equal manner. The process of designing algorithms for load balancing is performed by considering a number of factors, such as the capabilities that can be achieved, the architecture of the cloud infrastructure, the degree of network bandwidth, and the nature of the applications. Poorly managed instances of load balancing across cloud computing servers can degrade the performance of applications in real time. The factors of resource utilization, service response time, and cost reduction are vital for the performance improvement of cloud services. A cloud computing system should enhance the quality of service. A cloud-based system should scale to a large number of users to distribute the load in an environment. [7]

Cloud infrastructures are composed of large data centers containing a large number of heterogeneous computational resources. Load balancing can provide improved fault tolerance and resource usage. A cloud-based load balancing system should have supporting architecture for scalability, fault tolerance, efficient resource utilization, etc. Contentions among resources associated with load balancing can affect the performance of the cloud. Balancing the incoming requests by mapping them to the servers is the role of load balancing architectures. In a cloud computing system, based on heterogeneous algorithms, load balancing performs to avoid resource contention in physical systems. A cloud-based system receives requests. In the fault scenario, the request should be forwarded without interruption to the passive or available servers. The implementation of the supported resources should be efficient. In a cloud-based environment, the strategy for balancing the load should serve large datasets when millions of files are shared. [8]

### 4.1. Concepts and Principles

Cloud computing is the technology that enables users to access and consume machine-level resources as utilities via the Internet. Dynamically allocating resources is one of the approaches to achieve high quality of service. Load balancing is a crucial, ongoing task in cloud computing due to the dynamic and time-varying resource requirements of users and service request categories. It is the process of making decisions about the allocation of computer system resources so that the system works correctly and efficiently, as well as the distribution of the workloads to available resources. There are many factors to be considered, as follows: computing speed, storage capacity, and network bandwidth, which are resources to be allocated; the size, complexity, and urgency of tasks, which are the workloads to be offloaded; and the load distribution expectations assumed and the consumer temporal access pattern. [9]

10.48047/jocaaa.2024.33.07.21

Cloud resources and user workloads are not fixed relationships, due to dynamic system topology, diverse and time-varying user demands, and the distribution of workloads not being balanced. So dynamic load balancers have become more important to overcome this issue. Many studies have thrived on developing various algorithms for load balancing, including genetic algorithms, artificial neural networks, shop algorithms, k-means algorithms, particle swarm optimization, ant colony optimization methods, and several hybrid algorithms. Virtualization has progressed in consolidating application virtual machines onto physical servers and provides independent run-time execution environments for different end-user applications based on hypervisors. Elasticity is the logical scaling of on-demand computing resources upon request, namely providing additional capacities that were not previously available in advance. Scalability is the system's ability to handle increasing system load and is a measure of how well a cloud infrastructure is coping with the increasing demand for its services. [10]

Dynamic load balancing is a system of calling load balancing that does not consider resources and computing at the moment but needs to expand the characteristics of cloud computing under the situation of cloud services operating; it also manages and arranges the running resources of the system according to the operational status of jobs in the state of load management, and alters the system elements such as load balancer and resource programming within a certain range. Native and other identifiers are derived from dealing with testing and scheduling arrangements.

### 4.2. Challenges and Issues

In cloud computing environments, load balancing (LB) refers to the distribution of workload across multiple computers, networks, or other resources to achieve the optimal level of resource utilization, avoid unexpected congestion, and reduce response time. This is especially challenging if the real-time workloads are highly dynamic and the user demands are imprecise. The cloud computing infrastructures are composed of thousands to tens of thousands of cores, numerous storage devices, and high-speed network components in a single data center, which need to manage, handle, and balance these resources effectively in real-time. A single resource failure among these system components can disrupt the services or have a huge impact on system performance. In addition to these challenges, the existing literature on the topic claims to balance the load of the resources, but the algorithms proposed for it cause high overhead, communication complexity, or introduce mutual exchange. Due to the high complexity, the solutions are not efficient for techniques such as message passing and computational overheads, and thus need more sophisticated load balancing algorithms to focus on this issue. The problem becomes even more complex in large-scale cloud infrastructure since the application and the hardware it runs on are non-homogeneous, and the bandwidth of the high-speed networks is reduced. We also have to ensure security and fault tolerance because these are the biggest and most important determining factors, which can lead to disturbances in selecting an appropriate load. In other words, acquiring legitimate or illegitimate user requests from 50 to 1000 transactions per second results in more than 1 million transactions per second, and ensuring an equal distribution of these requests in a legitimate or illegitimate manner in the network is likewise a difficult challenge that needs to be addressed. On the other hand, when the number of resources increases, such a distribution approach may adversely affect the performance of cloud computing environments. Given these

10.48047/jocaaa.2024.33.07.21

aspects, the algorithms or mechanisms that do any type of distribution should adopt the right approach by considering these aspects, which are challenging problems. [11]

**5. Hybrid Resource Utilization Load Balancing Algorithm**

The load balancing algorithm proposed in this paper is based on hybrid resource utilization. The algorithm is discussed in the context of the cloud-based leadership supercomputing system, although the generic requirements of the algorithm have been discussed wherever necessary. The hybrid resource utilization load balancing is made with the trade-off between performance and resource utilization. The resource allocation decision has been made with the concern of user requirements, while the administrators can also intervene if required. The requirement for security in the cloud has also been considered while designing the system for any new user/job that has been entered. [12]

The proposed algorithm has been designed from scratch. The architecture of the proposed algorithm has been discussed with design notes wherever required. The algorithm is equipped with innovative algorithmic techniques and computing paradigms such as stochastic concerns, game theory, and queuing-based solutions wherever required. Since the load may contain static as well as dynamic components, the design of the algorithm must consider the requirements of both dynamic and static components. The proposed hybrid resource utilization load balancing algorithm has been designed to balance the trade-off between users requiring bulk processing and soft QoS. The approach described previously can be scaled into various modern cloud dilemmas such as data streaming and grid computing environments with fewer tweaks. Furthermore, users have the requirement to parallelize some tasks but require them to be executed in a sequential order. This temporally depends on their results; this property is known as the dependency property of the jobs. This phenomenon can also be discussed under the provided infrastructure of the algorithm. Some of the major issues include processing deadline-related jobs, task dependencies, and resource utilization in a way that makes them available after completion. The major trade-offs include balancing resource utilization, existing resources, and other major factors that make this work attractive. The design of the algorithm is such that it makes a trade-off between main performance and accurate resource utilization. This trade-off is in line with current global execution systems, as computing timely and correct output is very important in addition to efficient resource utilization. Some industries may not be interested in getting results quickly but are interested in saving resources, while a major percentage of industries are interested in obtaining results in the fastest possible time, along with resource efficiency. Hence, the trade-off between these two major constraints has been addressed in the working algorithm. The design of the algorithm is based on experimental data, and we can evaluate the performance of the designed algorithm presented theoretically as well as experimentally. This will be done under real simulations of the processor as well as the scheduler. In our architecture, resource management is based on criteria that can be considered and viewed as resource allocation. If two users exist in the provided architecture, then the earlier user will be allocated the job before the later one. Furthermore, if two users exist and the system has the capability to process and allocate jobs for both users, then the time parameters with the assigned jobs have also been maintained. Moreover,

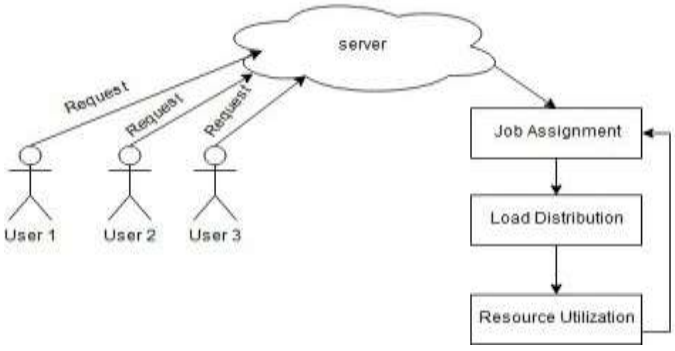
10.48047/jocaaa.2024.33.07.21

the given job will access processing only if its prerequisites have been completed. The proposed schedule enqueues or dequeues it if waiting. [13][14]

### 5.1. Design and Architecture

The working model of HRULA mainly consists of three modules: the resource management module, the flow scheduling module, and the resource status collecting module. In the flow scheduling module, we present the structural framework and workflows of the HRULA model. The workflow of the HRULA-based resource allocation proceeds through four steps: 1) resource requirement estimations, 2) flow scheduling based on the weighted least connections and weighted requested service rate algorithm, 3) resource status collecting and analyzing, and 4) task migration. The rectangles in the graph are parts of a process, and the ovals are data. All labels indicate the operation of each individual part of the process required to satisfy the functional description.

The first step of this process is estimating the resources a flow needs in advance, and the estimated resource utilization can be calculated. Therefore, we can allocate the flow's average bandwidth requirement, flow's arrival rate, and average service rates to a hybrid servicer module. In the process, when the flow scheduling module schedules service rate servers, after a time interval, the system resources will be reconfigured if we find that the system resources cannot meet the new arriving flow's servicer rate demand. In short, HRULA integrates the virtues of the two well-known load balancing mechanisms: weighted least connections and the weighted requested service rate mechanisms. The load balancer uses the queue length and running tasks as respective load metrics. Balancing the arriving requests among the working servers helps achieve high performance, such as decreased request times, increased number of requests handled, and decreased number of long waiting times. As a consequence, considering the practicability of the algorithm, HRULA is supposed to fit a variety of different cloud environments and is optimal in a heterogeneous circumstance.




Proposed System Model

### 5.2. Key Components

The Hybrid Resource Utilization Load Balancing Algorithm is divided into four chief components: Manager, Balancer, Predictor, and Quality Manager. It is integral to understand the

10.48047/jocaaa.2024.33.07.21

functions of these components in detail to comprehend the Hybrid Resource Utilization Load Balancing Algorithm.

Comparison Among Various Job Assignment Techniques

| No. of Tasks | Sequential | Linear Hashing | Quadratic Hashing |
|---|---|---|---|
| 10 | .72 | 1.06 | .71 |
| 50 | 1.46 | 2.43 | 1.23 |
| 100 | 2.41 | 3.92 | 2.12 |
| 200 | 3.5 | 4.25 | 2.91 |
| 400 | 5.8 | 6.79 | 4.5 |

According to the aforementioned table, the quadratic probing methodology for hashing has been selected for job assignment due to its superior performance relative to other methods.

Comparison of Average Makespan

| Experiments (Tasks/VM) | LBDA | Proposed Algorithm |
|---|---|---|
| 100/6 | 360 | 161.96 |
| 150/8 | 427 | 205.02 |
| 200/10 | 430 | 277.34 |
| 250/12 | 433 | 302.67 |
| 300/14 | 458 | 319.59 |

The findings presented in Table indicate that the suggested method outperforms LBDA regarding minimal average makespan. The results are graphically shown in Figure for enhanced comprehension.

A cloud environment incorporates the correlation and interaction of diverse entities to optimize resource utilization dynamically. Therefore, this paper presents novel techniques to manage an increasing amount of computing entities dynamically in a cloud architecture comprising public and private clouds and their respective workloads and resources; the basic functions of each are as follows: Manager: initiates communication with Hybrid RULBA services; contains information for each Balancer available in the hybrid cloud architecture. Balancer: handles the load requests to make intelligent decisions, accommodating Quality and Predictor Manager's

10.48047/jocaaa.2024.33.07.21

answers. It also communicates among balancers in order to handle periodic information requests. Predictor: performs this service for each prediction request by a Balancer and then sends the results back to the Balancer. In addition, it communicates with other Predictors in the private-public cloud architecture. Quality Manager: assesses reports to the Balancer, analyzes the quality of surplus workload distribution, and resource allocation. A large number of predominant challenges exist in developing and implementing an effective algorithm. Uplifting performance to exploit benefits in each tier is the main objective of the proposed model. The proposed four components interact to accomplish this. The chief function of each of the four components of the algorithm is discussed below.

## 6. Experimental Evaluation

We evaluate the performance of our cloud-based hybrid resource utilization load balancing algorithm using both simulation and real-world scenarios. In simulation, fictitious resource requests are submitted to this algorithm, and the time spent servicing each request and response time are recorded. In real-world scenarios, the Service Template system test bed is employed to distribute job submissions to imitate the multicloud. Messages written to log attributes vary according to the location of the user at submission time. We evaluate and compare the performance of both HRLB and hybrid using some statistical metrics such as response time and resource utilization.

The summary of results for hybrid, HRLB, and original is presented. It is observed that, on average, the highly loaded servers are 43% and 48% fully utilized by hybrid and HRLB algorithm respectively, compared to the 34% fully utilized by the original. This is an indication that the new algorithm is an improvement over the existing resource management algorithm. Results from the evaluation of the submission response time show that HRLB significantly improved the average submission response time by 39.65% and 42.64% over the original in simulation and real-world scenarios respectively. In the case of hybrid versus original, the average delay time is reduced by 34.17% and 35.91% in simulation and real-world scenarios. This improvement notwithstanding, the average submission response time is slightly better than that of hybrid by 7.75% and 6.25% in simulation and real-world scenarios. The evaluation of the algorithm showed that it exhibits good load balancing efficiency like the hybrid algorithm within these scenarios. Throughout the run of the experiments, latency is higher in the simulation scenario compared to the real-world scenario. This deviation can be attributed to the fact that the network is controlled by the time-sharing policy, in which during peak times, requests in the processing queue may be delayed. This is unlike the real-world scenario where the network handles incoming requests as received. The experiments were carried out using a few machines. The challenge encountered is that obtaining these machines at the same time is not practical, as the machines are in great demand; they were therefore run serially. Also, the internet is an untrusted medium where many variables can affect the performance of a configuration.

## 7. Conclusion and Future Directions

This paper proposes a cloud-based hybrid resource utilization load balancing algorithm to explore the optimal strategy from multi-layer relations and to maintain the global balance between the

10.48047/jocaaa.2024.33.07.21

inter-node and intra-node, as well as between CPU and memory usage. The designed performance metrics include minimizing the total waiting time for requests of the balancing mechanism, minimizing the nodes' sluggishness, maximizing the profit of hiring virtual resources, and minimizing the overhead of hiring and maintaining nodes.

In summary, this paper aims at both balancing the load and provisioning service levels. Despite making some assumptions regarding cloud computing characteristics, such as the cloud service flow process, we designed a hybrid strategy with global and local optimization to perform cloud-based load balancing. Our load balancing mechanism can play a vanguard role in the perspective of cloud computing. By adopting the hybrid distributed approach, a cloud provider can monitor the service level or quality of service in a proactive manner, and the results of hiring nodes can guarantee the performance and stability of each node and of the raw physical resources in the cloud. The cost is reduced by using the two types of jobs, resulting in a better overall return. Although we have tackled several open issues, we are entering a new territory to be explored.

In the near future, as computational, sensing, and networking technologies continue to innovate, this LR-LBM can be implemented on a massive scale, making the policy-driven IRM among the four components even more effective. We can indeed have an autonomic computing and communication environment. The LBM will be some linguistic version that can be formulated in mathematical forms and can be applied in many other fields as well, with minimal or no cost in hardware while yielding excellent performance. Regarding further work, we need to be able to provide performance for the overall latency and throughput of the cloud, which the provision of an allocation-based LR-LBM as part of the policy-driven IRM can realize and ensure.

References:

[1] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *Journal of King Saud University-Computer and Information Sciences*, 2022. sciencedirect.com

[2] N. Chauhan, N. Kaur, and K. S. Saini, "Performance Analysis of Rules Generated Hybrid Optimization Algorithm for Resource Allocation and Migration in the Cloud Environment," in 2023 2nd Edition of IEEE Delhi, 2023. [HTML]

[3] K. Geeta and V. Kamakshi Prasad, "Multi-objective cloud load-balancing with hybrid optimization," International Journal of Computers, 2023. [HTML]

[4] S. S. Sefati, M. Mousavinasab, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation," The Journal of …, 2022. researchgate.net

[5] P. K. Bal, S. K. Mohapatra, T. K. Das, K. Srinivasan et al., "A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques," Sensors, 2022. mdpi.com

10.48047/jocaaa.2024.33.07.21

[6] S. A. Bello, L. O. Oyedele, O. O. Akinade, and M. Bilal, "Cloud computing in construction industry: Use cases, benefits and challenges," Automation in Construction, 2021. sciencedirect.com

[7] M. Hamdan, E. Hassan, A. Abdelaziz, and A. Elhigazi, "A comprehensive survey of load balancing techniques in software-defined network," *Journal of Network and …*, 2021. academia.edu

[8] L. Helali and M. N. Omri, "A survey of data center consolidation in cloud computing systems," Computer Science Review, 2021. researchgate.net

[9] F. Alqahtani, M. Amoon, and A. A. Nasr, "Reliable scheduling and load balancing for requests in cloud-fog computing," Peer-to-Peer Networking and …, 2021. [HTML]

[10] N. M. Abdulkareem, "Optimization of Load Balancing Algorithms to Deal with DDoS Attacks Using Whale Optimization Algorithm," Journal of Duhok University, 2022. academia.edu

[11] J. Nazir, M. W. Iqbal, T. Alyas, and M. Hamid, "Load balancing framework for cross-region tasks in cloud computing," Computers, Materials, 2022. academia.edu

[12] B. Kruekaew and W. Kimpan, "Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning," IEEE Access, 2022. ieee.org

[13] A. Narwal, "Resource Utilization Based on Hybrid WOA-LOA Optimization with Credit Based Resource Aware Load Balancing and Scheduling Algorithm for Cloud Computing," Journal of Grid Computing, 2024. [HTML]

[14] Z. Nezami, K. Zamanifar, and K. Djemame, "Decentralized edge-to-cloud load balancing: Service placement for the Internet of Things," IEEE Access, 2021. ieee.org