TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52031

Load balancing and scheduling algorithms in cloud computing: A Review

Deepak Joshi[a], Kumar Bibhuti Bhushan Singh[b]

[a] Scholar, Department of Computer Science and Engineering, Goel Institute of Technology & Management, Lucknow, India

[b] Assistant Professor, Department of Computer Science and Engineering, Goel Institute of Technology & Management, Lucknow, India

KEYWORD

Cloud Computing, Static Algorithm, Dynamic Algorithm, Load Balancing, Task Scheduling, Virtual Machine

ABSTRACT

Cloud computing is a relatively new technological innovation that has brought about a revolution in how people, firms, and organizations utilize computer services. The use of cloud computing allows individuals and firms to access computing power via the internet whenever required. This can be in the form of storage capacity, software, databases, networking, and computational power. In recent times, cloud computing has emerged as the most important innovation in the field of information technology due to its ability to provide flexible and faster computing services. With the availability of computing services via the internet, users can now perform various tasks using any internet-enabled device, including their laptop computers and smartphones Cloud computing refers to offering services on demand. One does not have to buy costly machines or construct vast data centers since the cloud computing provider takes care of these aspects. Some companies that provide cloud services include Amazon Web Services, Microsoft, and Google. Cloud computing enables firms to increase their activities as per their requirements. Cloud computing is common in education, healthcare, financial institutions, entertainment, and many other industries. For instance, virtual classrooms, video streaming software, and online payment services require cloud computing extensively.

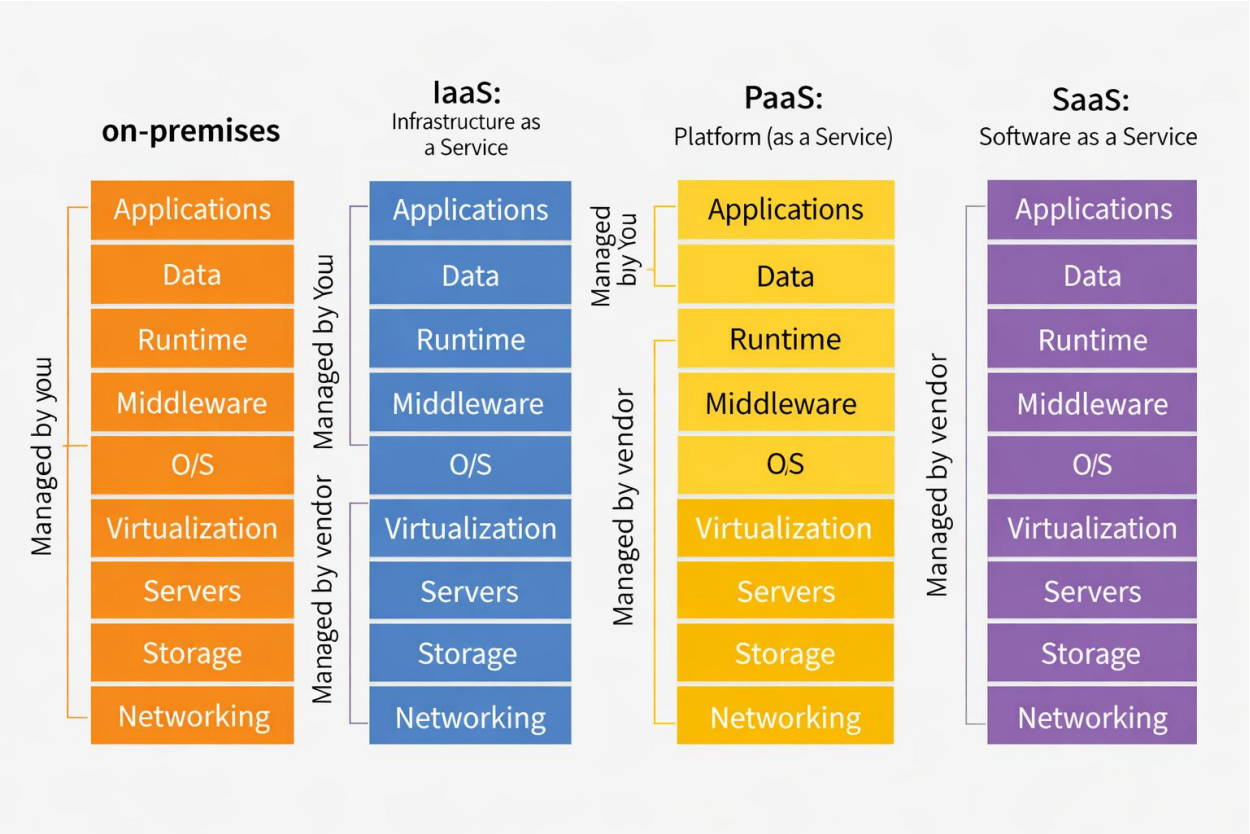
I. INTRODUCTION

Cloud computing refers to offering services on demand. One does not have to buy costly machines or construct vast data centers since the cloud computing provider takes care of these aspects. Some companies that provide cloud services include Amazon Web Services, Microsoft, and Google [1]. Cloud computing enables firms to increase their activities as per their requirements. Cloud computing is common in education, healthcare, financial institutions, entertainment, and many other industries [2]. For instance, virtual classrooms, video streaming software, and online payment services require cloud computing extensively [3].

Corresponding Author: Kumar Bibhuti Bhushan Singh, Assistant Professor, Department of Computer Science and Engineering, Goel Institute of Technology & Management, Lucknow, India
Email: k.bibhuti.bhushan@goel.edu.in

II. CLOUD COMPUTING

Cloud computing is a pay-per-use service that gives you access to software, hardware, and infrastructure [4]. It lets users get scalable resources like power as needed, so they don't have to be watched over by people. Cloud computing utilizes globally distributed data centers to bypass numerous financial and physical obstacles to IT services [5]. It has quickly emerged as a catalyst for innovation in multiple sectors by guaranteeing adherence to Service Level Agreements (SLA) and upholding Quality of Service (QoS) [6].



III. LOAD BALANCING

Load balancing within cloud infrastructures involves evenly allocating tasks among multiple computing resources, such as servers, VMs, or hosts—ensuring efficiency, reliability, and balanced utilization [7]. It is a very important method that prevents any single resource from being overloaded and surrounded with too many requests while others remain underutilized or used much less than expected [8]. By smartly managing the allocation or distributing of tasks, load balancing improves system performance, reduces response times, and improves throughput, making sure that users experience consistent and reliable services [9]. Basically, load balancing acts as the "traffic controller" of cloud environments, directing workloads to the good, efficient resources, depending on availability and capacity to handle load[10]. The functioning of load balancing comprises supervising incoming requests and dynamically changing tasks, assigning them to resources [11].

Aspect	**Load Balancing Algorithms**	**Scheduling Algorithms**
Primary Goal	Distribute workloads evenly across resources	Decide execution order and allocate tasks to resources
Focus	Resource-level (e.g., servers, VMs)	Task-level (e.g., cloudlets, jobs)
Key Benefits	Prevent overload, improve reliability and availability	Optimize resource usage, reduce makespan, increase throughput
Adaptability	Static (e.g., Round Robin) vs Dynamic (e.g., ACO, Honeybee)	Static vs Dynamic, including Bio-inspired approaches

Metrics Used	Response time, throughput, resource utilization	Makespan, throughput, fairness, resource usage
Impact	Enhances system stability, scalability, and ability to handle faults	Improves task execution efficiency and system performance

Comparison of Load Balancing and Scheduling Algorithms in Cloud Computing

IV. OBJECTIVES OF LOAD BALANCING

The most important goal of load balancing in cloud computing is to make sure that workloads are distributed efficiently across available resources, to maximize performance and minimize delays in service.

V. MOTIVATION

With the increasing growth of data, applications, and services hosted in the cloud, efficient resource management has become a critical challenge. Many traditional static algorithms are not flexible enough to accommodate changes in demand fluctuations, resulting in inefficiencies, delayed responses, and misuse of both VMs and their underlying physical host machines. This constraint makes it essential to have flexibility in order to achieve enhanced scheduling and load balancing techniques that can dynamically adapt to changing workload profiles.

VI. EXISTING LOAD BALANCING AND SCHEDULING ALGORITHMS

Load balancing in cloud computing has been widely explored, with numerous algorithms developed to tackle the challenges of efficiently distributing workloads across available resources [12]. These algorithms can be broadly categorized into static and dynamic methods; each method has its own strengths and limitations [13]. There several algorithms which are designed for different purposes some examples are Round Robin algorithm, Weighted Round Robin Algorithm, Least Connections Algorithm, Shortest Response Time Algorithm, Throttled Load Balancing, Honeybee Foraging, Ant Colony Optimization, Genetic Algorithms and Particle Swarm Optimization [14].

Hardware-Based Load Balancing: It's a dedicated physical infrastructure prepared with a special operating system designed to distribute workloads in a way that produces a lot of efficiency among servers . All incoming requests are first directed to the load balancer, which then decides the best fit server to handle each request . Hardware-based solutions offer high performance and reliability, with very little chances of failure . However, they involve significant installation costs, require added IT infrastructure, and may cause extra expenses during peak workload periods .

Software-Based Load Balancing: Software-based load balancing distributes workloads across servers using predefined or able to change and get better, algorithms . This approach is more productive in terms of cost since it does not require added hardware only the development and deployment of best fit algorithms . It can automatically dynamically supply resources on demand . Although more flexible and money saving, its performance is generally lower compared to hardware-based load balancing .

VII. STATIC ALGORITHMS

Static algorithms operate on predefined rules and do not change according to real time changes in workload or resource conditions [15]. One of the simplest static methods is the Round Robin algorithm, which assigns tasks sequentially, one after the other to available servers in a circular order [16]. This makes sure of fairness but does not take care of current load or capacity of each server. A different version of this is Weighted Round Robin algorithm, where servers are assigned, weights based on their capacity, ability, and tasks are distributed accordingly in the same way [17]. Another static approach is the Randomized algorithm, which assigns tasks randomly to servers, reducing the chance of bottlenecks but missing to predict a possible future event and its ability [18]. While static algorithms are easy to use and require very little overhead, they are less effective in dynamic changing environments where workloads go up and down significantly [19].

VIII. DYNAMIC ALGORITHMS

Dynamic algorithms continuously monitor system performance and change to make better fit for new conditions taking care job are distributed according to current situation [20]. These are hard to implement but provide better ability to change and efficiency. One widely used dynamic, changing method is Least Connection

algorithm, which move tasks to the server with have fewest active connections, making sure of balanced resources usage [21]. In almost the same way, the Shortest Response Time algorithm assigns tasks to servers that respond fastest, improving user experience [22]. Another dynamic, changing approach is the Throttled algorithm, controlled the speed of a request given, where each server is assigned a limit, and tasks are given out based on availability within those limits [23]. Dynamic algorithms also include bio inspired ways of doing operation, which copy natural processes to accomplish smart load distribution [24]. The Honey Bee Foraging algorithm models the behaviour of bees searching for food, where tasks are given out based on profit values, directing workloads toward servers that demonstrate better performance [25].

IX. METHODOLOGY

The experiment begins with the initialization of the CloudSim environment, where a datacenter is created to represent the physical infrastructure, and basic conditions for experiments. This datacenter consists of a host prepared with multiple processing elements (PEs), RAM, bandwidth, and valuable storage resources. On top of this infrastructure, five heterogeneous virtual machines (VMs) of different capacities are deployed, each configured with different MIPS values to test servers of different capacities. The workload is modeled using fifty cloudlets, representing user tasks with irregular job sizes. After execution, the simulation, it collects key performance metrics which measures like average response time, throughput, and utilization. As part of our CloudSim experiments, we conducted task scheduling and load balancing evaluations at two distinct levels, namely the VM level and the Host level, to capture performance variations across both layers.

X. RESULT AND DISCUSSION

We systematically analyze the data obtained from both scenarios and interpret the results to identify key trends, variations, and performance improvements.

Algorithm	Average Response Time(seconds)	Throughput (cloudlets/sec)	VM Utilization Ratio (0–1)
Round Robin Algorithm	141.7754	0.0800	0.1046
Weighted Round Robin Algorithm	57.3581	0.1648	0.2044
Honey Bee Foraging Algorithm (bio-inspired)	43.1352	0.2189	0.2655
Ant Colony Optimization Algorithm (Bio-Inspired)	36.7168	0.3822	0.4057

Table 1: Performance Metrics (Simulation Case 1)

Algorithm	**Average Response (seconds)**	**Throughput (cloudlets/second)**	**VM Utilization Ratio (0-1)**	**Host Utilization Ratio (0-1)**
Round Robin Algorithm	151.1064	0.0778	0.1121	0.0249
Weighted Round Robin Algorithm	52.0253	0.1723	0.2060	0.0458
Honey Bee Foraging Algorithm	44.4310	0.2671	0.2952	0.0656
Ant Colony Optimization Algorithm	19.5147	0.2454	0.3535	0.0786

Table 2: Performance Metrics (Simulation Case 2)

When we examine the performance metrics table for VM level load balancing and scheduling, the comparative and main differences between the four algorithms become very clear. Round Robin scheduling, which is the simplest and most traditional approach, shows the weakest performance with an average response time of 141.77 seconds, throughput of only 0.08 cloudlets per second, and a utilization ratio of resources 0.1046. These numbers highlight that while Round Robin makes sure of fairness by distributing tasks sequentially, one after the other across VMs, it fails to properly use the mixed-up nature of VM capacities. Heavy tasks are often assigned to weaker VMs, which increases waiting times and lowers overall system productivity. In contrast, Weighted Round Robin improves very much by incorporating static weights based on VM capacity. Its average response time drops to 57.35 seconds, throughput doubles to 0.1648 cloudlets per second, and utilization rises to 0.2044. This demonstrates that thinking about VM strength in scheduling decisions leads to better efficiency, though the static nature of weights means the algorithm cannot adapt dynamically, unable to change quickly as needed.

XI. CONCLUSION

The comparison of the simulation output from both the cases of simulation shows that the variation lies solely because of the adaptability feature. While the static algorithm acts as the basis, it is essentially limited when it comes to dealing with situations where the workloads are inconsistent and there is diversity in the nature of resources. The bio-inspired algorithms fit in well within such a scenario as they are specifically designed for such a scenario and continue to grow along with it. They do not follow any strict principles, but rather continue to adapt according to their real time conditions.

XII. FUTURE SCOPE

There are many promising areas that could be explored further based on the results of the current research and thesis paper in the sphere of cloud-related load balancing and scheduling issues. Comparisons of various approaches revealed that such nature-based algorithms as Honey Bee Foraging and Ant Colony Optimization provide greater efficiency compared to the existing conventional approaches. Nevertheless, there is still much work to be done to improve the performance of these algorithms further. The first thing one could consider is the implementation of such algorithms with emerging technologies in cloud computing, such as container orchestration services and edge computing technologies. Given the increasing tendency towards using microservices architecture and edge computing solutions, future scheduling technologies should become more sophisticated. Hence, a deep analysis of the possibilities for the implementation of Honey Bee and Ant Colony approaches to container and edge node scheduling should reveal their actual capabilities.

The development of hybrid algorithms is another field that should be considered for further investigation. Although bio-inspired algorithms exhibit great flexibility, their integration with prediction tools like machine learning algorithms would result in even greater efficiency. The use of reinforcement learning in optimizing the decision structure of Ant Colony Optimization, or the implementation of prediction algorithms in managing profit updates in Honey Bee Foraging algorithms, are some possible solutions. Such hybrids would allow for adaptation in terms of both events and workload predictions. Such an approach to applying the heuristic techniques derived from biology and based on predictive analysis could provide a highly interesting new area for studying cloud computing tasks scheduling algorithms. Also, in future research, these algorithms should be studied in more general environments, where their performance would be assessed in case of nonuniform workloads. At the moment, all the tests involved irregular workloads, but real cloud-based computing systems can comprise different tenants, multiple geographically distant datacenters, and various levels of service-level agreement contracts. Such tests would offer additional insights regarding the behavior of proposed solutions. Furthermore, the evaluation of the influence of environmental parameters on the performance of these algorithms could link them with the emerging area of green computing. Indeed, bio-inspired heuristics, thanks to their adaptability and flexibility, would be able to achieve high performance levels while maintaining energy efficiency.

REFERENCES

[1]. R. Buyya et al., Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.

[2]. P. Mell and T. Grance, NIST Special Publication 800-145, 2011.

[3]. M. Armbrust et al., Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[4]. L. Vaquero et al., ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50–55, 2009.

[5]. E. Caron et al., in Proc. IEEE Cloud Computing, 2010, pp. 456–463.

[6]. J. Broberg et al., Journal of Grid Computing, vol. 6, no. 3, pp. 255–276, 2008.

[7]. M. Randles et al., in Proc. IEEE AINA Workshops, 2010, pp. 551–556.

[8] A. Khiyaita et al., in Proc. IEEE Cloud Computing Technologies and Applications, 2012, pp. 106–111.

[9]. Y. Fang et al., Journal of Computers, vol. 6, no. 1, pp. 132–139, 2011.

[10]. A. Beloglazov and R. Buyya, Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp. 1397–1420, 2012.

[11]. J. Xu et al., in Proc. IEEE MASCOTS, 2007, pp. 327–334.

[12]. H. Liu et al., in Proc. IEEE ICPADS, 2011, pp. 9–16.

[13]. T. Kokilavani and D. Amalarethinam, International Journal of Computer Applications, vol. 37, no. 3, pp. 11–16, 2012.

[14]. S. Wang et al., Journal of Computers, vol. 8, no. 5, pp. 1315–1322, 2013.

[15]. M. Dorigo and T. Stützle, Ant Colony Optimization. MIT Press, 2004.

[16]. D. Karaboga and B. Basturk, Journal of Global Optimization, vol. 39, no. 3, pp. 459–471, 2007.

[17]. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

[18] . J. Kennedy and R. Eberhart, in Proc. IEEE Int. Conf. Neural Networks, vol. 4, 1995, pp. 1942–1948.

[19]. S. Chhabra and R. Singh, International Journal of Computer Applications, vol. 117, no. 18, pp. 5–8, 2015.

[20]. S. D. Kamal et al., International Journal of Cloud Computing and Services Science, vol. 2, no. 2, pp. 118–127, 2013.

[21]. R. N. Calheiros et al., "CloudSim: A toolkit for modeling and simulation of cloud computing environments," Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.

[22]. K. Li et al., "Efficient resource management for cloud computing," Future Generation Computer Systems, vol. 29, no. 1, pp. 1–12, 2013.

[23]. H. Zhang and G. Guo, "Load balancing in cloud computing: A survey," Journal of Computer Science and Technology, vol. 27, no. 2, pp. 356–371, 2012.

[24]. N. S. Gill and S. Singh, "A comparative study of load balancing algorithms in cloud computing," International Journal of Computer Applications, vol. 84, no. 12, pp. 1–4, 2013.

[25]. R. Kaur and P. Kaur, "Bio-inspired load balancing techniques in cloud computing," International Journal of Computer Applications, vol. 975, no. 8887, pp. 12–18, 2015.