International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Optimal Dynamic Load Balancing for Cloud Task Distribution Using Bayesian Model

[1.]T Kannadasan, [2]Dr. R. Pragaladan, [3]N. Sureshbabu,

Submitted: 06/02/2024 **Revised**: 14/03/2024 **Accepted**: 20/03/2024

Abstract- Cloud computing offers users on-demand services through internet on the basis of pay-per-use model. Due to the increased user, the need for resource sharing and utilization is growing quickly, which presents many difficulties for cloud computing. One of the most crucial aspects of cloud computing is load balancing, which evenly distributes the workload on available resources to avoid overloaded or under-load situation and improve the resource utilization. This paper suggests an optimal dynamic load balancing for task distribution problem in cloud environment using Bayesian Model called as LBBM. The two main components of this model are Load Balancer (LdBr) and Virtual Machine Monitor (VMMr). The LdBr assigns user tasks between available VMs and the VMMr analyze the available VMs and send the current status of each VMs to LdBr for task distribution. This approach optimally allocate task to the selected VM using Bayesian Model which reduces the makespan and increase resource utilization. The proposed LBBM approach is simulated using CloudSim Simulator. Simulation findings clearly demonstrate that the suggested strategy outperforms previous approaches in terms of lowering makespan and improving resource consumption.

Keywords: *Cloud computing, load balancing, task allocation, optimization, Bayesian model*

1. Introduction

Using a pay-per-use business model, cloud computing allows client to utilize a shared collection of resources, including computation, network, storage, and applications, on demand. In general, end users can access cloud computing resources through three various service kinds, including infrastructure as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS) [1]. In cloud each data center has a number of hosts, and each host can contain at least one virtual machine (VM) to allow customers to run applications without restriction.

In the cloud, load balancing offers methods for effectively distributing the workload (task) across all nodes (VM). The successful implementation of load balancing influences fail-over, boosts system adaptability, avoids system inefficiencies, and speeds up execution [2]. To obtain optimum accessibility, cloud providers have used load balancing strategies. During execution of tasks, methodologies for dynamic cloud load balancing disperse given tasks to virtual computers, and these virtual computers' load is changed based on the system's condition. Load balancing eliminates resource overload and decrease the overall waiting time for resources [3]. Figure 1 shows the load balancing model.

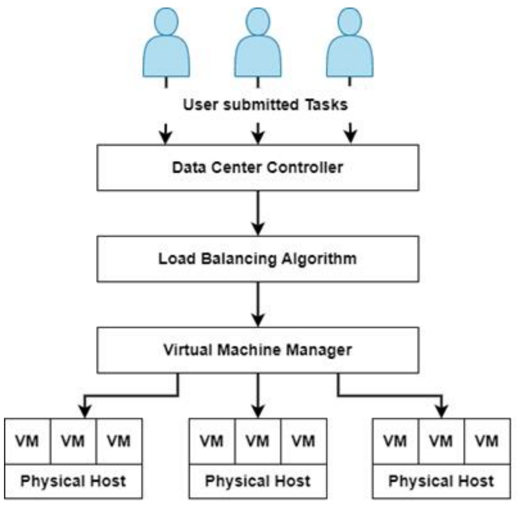

Figure 1 Load Balancing Model

Cloud load balancing techniques are classified into three types [4]: static, dynamic and hybrid. The static cloud load balancing methods [5][6] disperse the tasks among virtual hosts before the processing time starts. These methods waste resources by loading some virtual hosts during execution. Static load balancing techniques enhance resource expenses and raise SLA breaches caused by overloaded virtual hosts. Dynamic cloud load balancing techniques [7][8] allocate submitted tasks to virtual machines during task execution as a load of VMs is adjusted based on the system's condition. In dynamic load balancing, historic data about system state is ignored; this can overcome the shortcomings of static techniques. Hybrid algorithms [9][10] are a fusion of

[1]*Associate Professor of Computer Science, Thanthai Periyar Government Arts and Science College (Autonomous)*
profkannadasant@gmail.com
[2]*Assistant Professor & Head, Department of Computer Science, Sri Vasav College, Erode,*
pragaladanr@gmail.com
[3]*Assistant Professor, PG and Research Department of Computer Science Rajah Serfoji Government College, Thanjavur*
sureshbabunagarajan@yahoo.co.in

elements from dynamic and static cloud load balancing techniques.

The load balancing method is further divided into three categories: VM, task and resource load balancing. The VM load balancing [11] approach allocates VMs from heavily loaded to under load. The task load balancing [12] method allocates task uniformly between VMs. The resource load balancing [13] techniques regulate available resources. In cloud computing, dynamic task, VM load balancing is most crucial components of scheduling. The workloads must be split among numerous VMs in order to decrease response times and makespan. An effective load balancing method minimizes overuse or even depletion of the resources that are available.

This work proposes an optimal dynamic load balancing for task distribution problems in a cloud context using the LBBM Bayesian Model. The Load Balancer (LdBr) and Virtual Machine Monitor (VMMr) are the two key components of this approach. The LdBr distributes user tasks among available VMs, while the VMMr analyses the available VMs and sends the current status of each VM to the LdBr for task distribution. This solution uses a Bayesian Model to optimally allocate tasks to the selected VM, reducing the makespan and increasing resource utilization.

The remaining part of this paper is arranged as follows: Section 2 provides the related work of load balancing in cloud computing. Section 3 describes the formulation of proposed problem. The proposed task distribution with load balancing using Bayesian Model is explained in Section4. Section 5 discusses the experimental setup and results. Finally, section 5 concludes the paper with future direction.

2. Related Work

One of the most crucial aspects of cloud computing is load balancing, and numerous articles [14] [15] have addressed this problem. Souravlas et al., [16] proposes load balancing for task scheduling problem based on the Markov process model. The tasks are efficiently distributed among the VMs in the system. With this approach, each VM receives the workload in accordance to its available processing power. The implementation of this assignment assumes that, following the allocation of the new tasks, all VMs would be used equally.

A resource-aware dynamic task scheduling method is suggested by Nabi et al. in [17]. It load-balances a number of distinct, non-preemptive, and computationally demanding jobs onto the available resources. At runtime, it also modifies the load and compute share of the VMs. It uses a load-balanced task across the resources available to use all the machines evenly, maximizing resource consumption and obtaining the quickest task scheduling execution time.

Semmoud et al., [18] presented an adaptive starvation threshold-based task scheduling and load balancing solution for cloud computing. The threshold is frequently adjusted based on the volume of fulfilled requests and the duration of virtual machine idle time. To improve the Quality of Service (QoS) of Cloud systems, this technique also considers the activities' priority level. It increases system stability by reducing the number of migrations, which reduces task execution time and virtual machine idle time.

Nabi et al., [19] developed resource aware time limited load balancing technique. It has the capability of allocating the workload of autonomous and computationally demanding tasks in accordance with the resource computation capacity during runtime. It improves load balancing, supports deadline-driven workloads, and boosts the Cloud's overall performance.

In order to decrease the load imbalance of VM and task rejection rate, Tong et al. [20] proposed unique deep reinforcement learning-based dynamic load balancing task scheduling technique. This technique chooses a VM that is appropriate for the task before assessing whether executing the task on the chosen VM violates the Service Level Agreement (SLA). If the SLA is broken, the job is rejected and feedback is given as a negative penalty. If not, the task is accepted, run, and feedback is given based on the load distribution among the VMs.

Gupta et al. [21] proposed a hyper heuristic approach that seeks to achieve globally adjusted load across virtual machines while reducing makespan. It uses honey bee load balancing to give balanced scheduling solutions. Vijarania [22] suggests genetic algorithm for load balanced task scheduling.

A dynamic load balancing approach was presented by Kumar et al., [23] to reduce the Makespan and effectively use resources. Using the bubble sort algorithm, it arranges tasks according to length and computing performance. Then, in a First-Come-First-Served fashion, tasks are assigned to VM. Following the completion of allocation, the load is balanced while accounting for the load placed on virtual machines. This method can quickly minimize Makespan and maximize the available resources, however it ignores priority or other QoS criteria, including Timeline.

In case of imbalance issues, Polepally et al., [24] proposed load balancing algorithm for reallocating tasks to VMs. The method determines the ideal threshold value using the Dragonfly optimization algorithm. This samples is used to compare and estimate the load on VMs.

3. Problem Formulation

When the user submits the task in cloud, the system will distribute the task to available VMs in the cloud data center. Generally, the workloads are assigned arbitrarily to the VMs. The task cannot be deployed in the VM, when the task request is more than the actual resource of VM. When the required resource level is near to the physical host's available quantity, the physical host will experience a significant burden, resulting in a decrease in service capacity and computing power. The decrease will cause a load imbalance in the cloud data centre and a degrade service efficiency [25]. The effective task deployment technique should be capable of improving the overall load balancing effect of the cloud computing system. In a cloud data centre, it is required to develop an effective and load-balancing task distribution method.

Cloud computing is comprised of a group of virtual machines (VMs), and each VM is in charge of allocating and balancing the load by distributing VMs to hosts during load balancing across all servers. The task distribution problem is formulated as follows: Let consider T = {T_1, T_2, T_3,…,Tm} be the set of m Task provided by the user and VM = {VM_1, VM_2, VM_3…,VM_n} be the set of m Virtual Machine in cloud data center. Each VM resources: Memory, CPU and bandwidth. The goal is to schedule all tasks to running virtual machines in such a way that cloud users can complete their tasks in the shortest amount of time while maximizing resource utilization, i.e. user is expected to minimize makespan while cloud service provider is expected to maximize resource utilization.

4. Proposed Methodology

This section explains the proposed dynamic load balancing for task distribution problem using Bayesian model (LBBM). The Load Balancer (LdBr) and Virtual Machine Monitor (VMMr) are the two key components of this approach. The LdBr distributes user tasks among available VMs, while the VMMr analyzes the available VMs and sends the current status of each VM to the LdBr for task distribution. This solution uses a Bayesian Model to optimally allocate tasks to the selected VM, reducing the makespan and increasing resource utilization. Figure 2 shows the proposed framework.

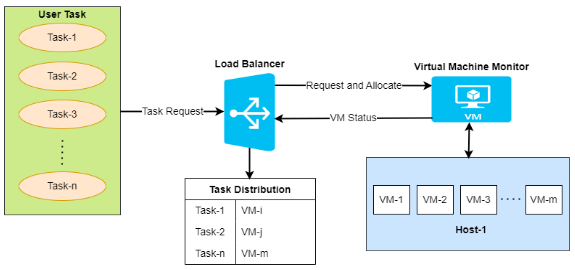

Figure 2 Proposed Architecture

Let consider T = {T_1, T_2, T_3,…,T_m} be the set of m Task provided by the user and VM = {VM_1, VM_2, VM_3…,VM_n} be the set of m Virtual Machine in cloud data center. Each task in T is represented by $T = \left(T_{id}, Req_{mi}^{i}, Req_{cpu}^{i}, 'Req_{mem}^{i}\right)$ where T_{id} is the task id, Req_{mi} is requested Million Instructions, Req_{cpu} is requested amount of CPU and Req_{mem} is requsted amount of memory. Each VM in list is represented by $VM = \left(VM_{id}, C_{mips}^{i}, C_{cpu}^{i}, 'C_{mem}^{i}\right)$ and C indicates the capacity of VM. The binary mapping matrix of task distribution among VM is denoted as,

$$TD_{mat} = \begin{pmatrix} td_{11} & td_{12} & \cdots & td_{1m} \\ \vdots & & \ddots & \vdots \\ td_{n1} & td_{n2} & \cdots & td_{nm} \end{pmatrix} \quad (1)$$

Here, n is the total number of Tasks, m is the total number of VM and td_{ij} = 1 if the task T_i is assigned to VM_j. Otherwise, td_{ij} = 0; Table 1 shows the symbols and notations.

Notation	Description
T_i	ith Task
VM_j	jth VM
$Req_{mi}^{i}, Req_{cpu}^{i}, 'Req_{mem}^{i}$	Million Instruction, CPU and memory (require) of ith VM
$C_{mips}^{i}, C_{cpu}^{i}, 'C_{mem}^{i}$	Million Instruction per second, CPU and memory (capacity) of jth VM
E_{ET}^{ij}	Expected execution time of ith task on jth VM
TET_i	Execution time of ith Task
RU_j	Resource utilization of jth VM
VM_{state}	State of VM
MS	Makespan
RU	Resource utilization

Table 1 Notation and Description

The excepted execution time of ith on jth is computed as follows:

$$E_{ET}^{ij} = \frac{Req_{mi}^{i}}{C_{mips}^{i}} \quad (2)$$

The execution time of ith task is calculated as follows:

$$TET_i = \sum_{j=1}^{n}\left(td_{ij} * \left(E_{ET}^{ij} + \frac{Req_{cpu}^{i}}{C_{cpu}^{i}} + \frac{Req_{mem}^{i}}{C_{mem}^{i}}\right)\right) \quad (3)$$

The makespan is computed as

$$MS = \max\{TET_i\} \quad (4)$$

The Resource utilization is calculated as

$$RU_j = \frac{Task\ Execution\ Time\ in\ VM_j}{MS} \quad (5)$$

The average resource utilization is computed as,

$$RU_{avg} = \frac{\sum_{j=1}^{n} RU_j}{n} \quad (6)$$

Each VM can be in one of three states: over utilized (OU), underutilized (UU), or normal utilized (NU). A virtual machine is UU if its utilization is less than 20% of its capability, and it is OU if its usage exceeds 80% of its capability. Otherwise, it remains NU, as shown in the following equation.

$$VM_{state}^{j} = \begin{cases} RU_j < 20\%, & UU \\ RU_j > 80\%, & OU \\ Otherwise, & NU \end{cases} \quad (7)$$

The VM is selected for task allocation based on the prior and posterior probability of each VM on the basis of Bayesian model. The prior probability of each VM is computed as,

$$P(X|VM_j) = 1 - \frac{Max\{Req_{cpu}^{i} * Req_{mem}^{i}\ ,\forall\ i=m\}}{Remaining\ Resource\ (VM_j)} \quad (8)$$

The posterior probability of VM is calculated as follows:

$$P(VM_j|X) = \frac{P(X|VM_j) * P(VM_j)}{\sum_{j=1}^{n} P(X|VM_j) * P(VM_j)} \quad (9)$$

Here, the $P(VM_j) = \frac{1}{n}$ (10)

Algorithm-1 explains the proposed dynamic load balancing for task distribution.

Algorithm-1 LBBM

Input: Task T = {T_1, T_2, T_3,…,T_m}, VM = {VM_1, VM_2, VM_3…,VM_n}

Output: Optimal Task Distribution TD_{mat}, MS, RU

```
Step01: For each i ∈ m
Step02:     For each j ∈ n
Step03:         Compute $E_{ET}^{ij}$ using (2)
Step04:     End For
Step05: End For
Step06: $T_1$ is randomly allocated to $VM_j$
Step07: While (Task $\neq \emptyset$)
Step08:     Find $VM_{state}^{j}$ using (7)
Step09:     Compute $P(VM_j|X)$ using (9)
Step10:     Sort $P(VM_j|X)$ in descending order
Step11:     For each j ∈ n
Step12:         if $T_i$ is suitable for $VM_j$ and $VM_j$ == UN or NU then
Step13:             Allocate $T_i$ to $VM_j$
Step14:             Break;
Step15:         End If
Step16:     End For
Step17:     Update $VM_{state}^{j}$
Step18: End While
Step19: Compute MS using (4)
Step20: Compute $RU_{avg}$ using (6)
```

5. Result and Discussion

This section explains the performance evaluation of the proposed LBBM. CloudSim 3.0 was used to simulate and analyse the suggested approach. It is an open-source toolset for simulating cloud computing environments. It was created in the CLOUDS lab at the University of Melbourne's department of computer and software engineering [14]. The CloudSim toolkit includes primary classes for defining data centres, virtual machines (VMs), applications, users, calculating resources, and policies for managing various system components (e.g., scheduling).

Type	Parameters	Value
DataCenter	Number of Data center	1
	Number of Hosts	1
Virtual Machine (Resources)	Number of VMs	5-100
	MIPS	500-5000
	VM memory(RAM)	512-4096 (MB)
	Number of PE per VM	1-8
Task	Number of Tasks	10-100

Table 2 Simulation Settings

The proposed work is evaluated through two metrics: Makespan and Resource utilization. This approach is compared with Souravlas et al., [16] (MPM) approach. Table 3 shows the makespan comparison.

No of Tasks	Makespan (Sec.)	
	MPM	LBBM
10	125	80
20	153	112
30	248	190
40	325	242
50	400	306

Table 3 Makespan Comparison

Figure 3 shows the comparison of makespan for different number of tasks. From that result the proposed approach has reduced makespan compared to MPM approach. When increasing the number of task the makespan also increased.

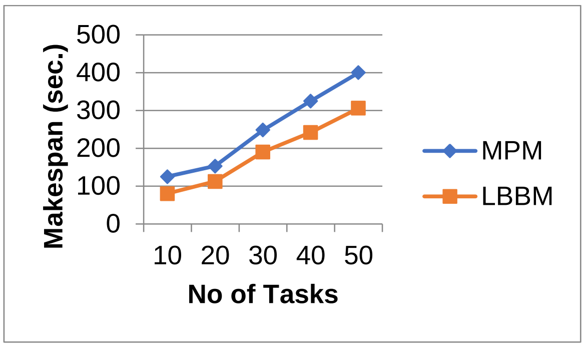

Figure 3 Comparison of Makespan for different number of Tasks

No of Tasks	Resource Utilization (%)	
	MPM	LBBM
10	60	72
20	65	79
30	69	82
40	72	85
50	75	89

Table 4 Resource Utilization Comparison

Figure 4 shows the comparison of resource utilization for different number of tasks. From that result the proposed approach has increased resource utilization compared to MPM approach.

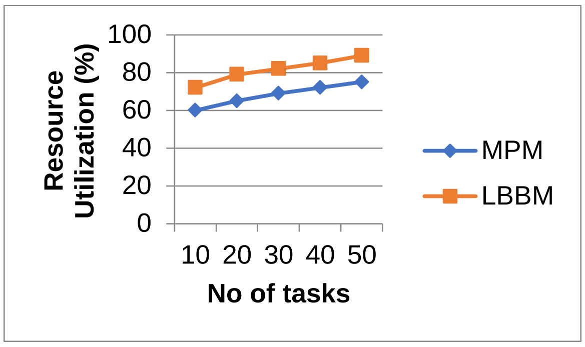

Figure 4 Comparison of Resource Utilization

Numerous load-balancing techniques have been proposed for distributed, grid, and cloud settings. However, each offers unique benefits and downsides. In the present investigation, a load balancing strategy based on the Bayesian model was used. The study's purpose was to adequately allocate the task among the resources so that none of them remained overloaded or underloaded. The status of each resource was calculated for this purpose, and load balance was maintained.

6. Conclusion and Future Work

Load balancing is critical and yet neglected in the cloud computing context. It is frequently often used an effective task scheduling mechanism to balance out the system's unbalanced load distribution. To properly use resources and minimize the makespan, an efficient load balancing solution is required. In relation to the aforementioned goal, this work presents dynamic load balancing using a Bayesian model. The two components Load balancer and Virtual machine monitor are efficiently distribute the tasks among VMs based on the VM current state and probability of VM usage. The simulation results shows that the proposed load balancing approach reduce makespan and increase resource utilization.

In the future, meta-heuristic-based solutions for the load balancing problem with task migration and VM selection could be addressed. This issue could be solved by introducing more tasks and virtual machines with hybrid model.

References

[1] Liu, F., Ma, Z., Wang, B., & Lin, W. (2019). A virtual machine consolidation algorithm based on ant colony system and extreme learning machine for cloud data center. IEEE Access, 8, 53-67.

[2] Junaid, M., Sohail, A., Rais, R. N. B., Ahmed, A., Khalid, O., Khan, I. A., et al., (2020). Modeling an optimized approach for load balancing in cloud. IEEE Access, 8, 173208-173226.

[3] Pradhan, A., Bisoy, S. K., & Mallick, P. K. (2020). Load balancing in cloud computing: Survey. In Innovation in Electrical Power Engineering,

Communication, and Computing Technology. Springer, 99-111.

[4] Tawfeeg, T. M., Yousif, A., Hassan, A., Alqhtani, S. M., Hamza, R., Bashir, M. B., & Ali, A. (2022). Cloud Dynamic Load Balancing and Reactive Fault Tolerance Techniques: A Systematic Literature Review (SLR). IEEE Access, 10.

[5] Shanthan, B.J.H., Arockiam, L., (2018). Resource based load balanced Min Min Algorithm for static meta task scheduling in cloud. IC-ACT'18, 1-5

[6] Miao, Z., Yong, P., Mei, Y., Quanjun, Y., & Xu, X. (2021). A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment. Future Generation Computer Systems, 115, 497-516.

[7] Liaqat, M., Naveed, A., Ali, R. L., Shuja, J., & Ko, K. M. (2019). Characterizing dynamic load balancing in cloud environments using virtual machine deployment models. IEEE Access, 7, 145767-145776.

[8] Hashem, W., Nashaat, H., & Rizk, R. (2017). Honey bee based load balancing in cloud computing. KSII Transactions on Internet and Information Systems (TIIS), 11(12), 5694-5711.

[9] Annie Poornima Princess, G., & Radhamani, A. S. (2021). A hybrid meta-heuristic for optimal load balancing in cloud computing. Journal of Grid Computing, 19(2), 1-22.

[10] Jena, U. K., Das, P. K., & Kabat, M. R. (2020). Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. Journal of King Saud University-Computer and Information Sciences, 34(6), 2332-2342

[11] Sui, X., Liu, D., Li, L., Wang, H., & Yang, H. (2019). Virtual machine scheduling strategy based on machine learning algorithms for load balancing. EURASIP Journal on Wireless Communications and Networking, 1-16.

[12] Ebadifard, F., Babamir, S. M., & Barani, S. (2020). A dynamic task scheduling algorithm improved by load balancing in cloud computing. In 2020 6th International Conference on Web Research (ICWR), IEEE, 177-183

[13] Kumar, J., Singh, A. K., & Mohan, A. (2021). Resource efficient load balancing framework for cloud data center networks. ETRI Journal, 43(1), 53-63.

[14] Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: a big picture. Journal of King Saud University-Computer and Information Sciences, 32(2), 149-158.

[15] Mohammadian, V., Navimipour, N. J., Hosseinzadeh, M., & Darwesh, A. (2021). Fault-tolerant load balancing in cloud computing: A systematic literature review. IEEE Access, 10, 12714-12731.

[16] Souravlas, S., Anastasiadou, S. D., Tantalaki, N., & Katsavounis, S. (2022). A Fair, Dynamic Load Balanced Task Distribution Strategy for Heterogeneous Cloud Platforms Based on Markov Process Modeling. IEEE Access, 10, 26149-26162.

[17] Nabi, S., Ibrahim, M., & Jimenez, J. M. (2021). DRALBA: dynamic and resource aware load balanced scheduling approach for cloud computing. IEEE Access, 9, 61283-61297.

[18] Semmoud, A., Hakem, M., Benmammar, B., & Charr, J. C. (2020). Load balancing in cloud computing environments based on adaptive starvation threshold. Concurrency and Computation: Practice and Experience, 32(11).

[19] Nabi, S., & Ahmed, M. (2021). OG-RADL: Overall performance-based resource-aware dynamic load-balancer for deadline constrained cloud tasks. The Journal of Supercomputing, 77(7), 7476-7508.s

[20] Tong, Z., Deng, X., Chen, H., & Mei, J. (2021). DDMTS: A novel dynamic load balancing scheduling scheme under SLA constraints in cloud computing. Journal of Parallel and Distributed Computing, 149, 138-148.

[21] Gupta, A., Bhadauria, H. S., & Singh, A. (2021). Load balancing based hyper heuristic algorithm for cloud task scheduling. Journal of Ambient Intelligence and Humanized Computing, 12(6), 5845-5852.

[22] Vijarania, M., Agrawal, A., & Sharma, M. M. (2021). Task Scheduling and Load Balancing Techniques Using Genetic Algorithm in Cloud Computing. In Soft Computing: Theories and Applications, Springer, 97-105

[23] Kumar, M., & Sharma, S. C. (2020). Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment. International Journal of Computers and Applications, 42(1), 108-117.

[24] Polepally, V., & Shahu Chatrapati, K. (2019). Dragonfly optimization and constraint measure-based load balancing in cloud computing. Cluster Computing, 22(1), 1099-1111.

[25] Zhao, J., Yang, K., Wei, X., Ding, Y., Hu, L., & Xu, G. (2015). A heuristic clustering-based task deployment approach for load balancing using Bayes theorem in cloud environment. IEEE Transactions on Parallel and Distributed Systems, 27(2), 305-316.