Enhancing Cloud Computing Efficiency with Crocodile Optimization Algorithm: A Novel Approach to Distributed Workload and VM Management

Ibrahim A. Ibrahim[1,*], Warshine Barry[2], Narek Badjajian[2]

[1]Computer Engineering, University of Technology, Baghdad, Iraq
[2]University of Debrecen, Department of Mathematical and Computational Science, Debrecen, Hungary

Email: ibrahim.a.almotairi@uotechnology.edu.iq; badjajiann6math@gmail.com; warshinabarrykurd@gmail.com

Abstract

Cloud computing has introduced itself as a mighty mechanism for delivering customers through the service model with on-demand, scalable, and instant access to computer resources. It will conduct effective load balancing and resource management, high importance so that the cloud system works with optimized performance and resource utilization. This gives a new strategy in load balancing and virtual machine (VM) control in cloud computing applied in the field using the Crocodile Optimization Algorithm (COA) for better performance. Inspired by crocodile hunting behaviors, the COA-based strategy is adopted to balance loads and manage VMs. This approach seeks to balance the number of the workload given to VMs with respect to the processing power of VMs and also the distribution of workload. It best uses resources in such a way that tasks are dynamically distributed to VMs in such a way that response time is at its minimum, and thus overall efficiency is enhanced in cloud computing. On the other hand, COA-based load balancing incorporates VM management techniques like migration and scaling to be adjustable in relation to the changing conditions of the workload. This allows dynamically adjusting the allocation of resources with respect to current demands, in such a way that assures optimal utilization of computational resources with high performance. The proposed approach was evaluated using simulations through CloudSim, one of the most adopted tools for simulating cloud computing. The COA effectively works are divided between the VM, which in turn will lead to better response time for the user request and improve cloud resource utilization. That is to mean, subsequent research would be some type of unique attempt in the area of load balancing and VM management in cloud computing, based on the Crocodile Optimization Algorithm. This approach improves efficient cloud computing through the balancing of load distribution, maximization of resource utilization, and lowering of response time.

Keywords: Crocodile Optimization Algorithm (COA); Distributed Workload; Cloud Computing; Virtual Machine (VM) Management; Resource Utilization

1. Introduction

Cloud computing has revolutionized sharing and using computing resources. It scales up the capability of acquiring resources. This, along with effective utilization of these resources and load balancing, plays a significant role in ensuring the best performance, hence benefiting from cloud systems [1]. This paper will unveil a novel approach recommended for managing Virtual Machines (VMs) in cloud computing and effective load balancing using the Crocodile Optimization Algorithm (COA). Essentially, in cloud computing, load balancing is a scenario where tasks are distributed across many VMs so that the likelihood of encountering a resource bottleneck is minimized, and an even distribution of resources can be achieved. Most traditional load balancing algorithms have been challenged in finding this balance, taking a step further to address these issues. Keeping the above challenges in mind, the paper discusses the proposed approach that leverages COA inspired by crocodile hunting. The load balancing approach based on COA dynamically allocates tasks to VMs, keeping in consideration their processing capability and current load. It generally aims at the balance in the distribution of workload over cloud infrastructure

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

and minimization in response time, along with maximizing resource utilization. It is a dynamic approach that can adapt to different demand conditions by executing VM task assignments for guaranteeing the efficient use of computing resources. In addition, the proposed system here includes VM management features: migration and scaling. This will enable dynamic adjustment of resource allocation on current demand, effectively utilizing computational resources while maintaining high performance. VM migration allows for the smooth transfer of VMs between physical servers, while scaling gives room for scaling VM capacity. This, therefore, calls for a need to discuss, present, and examine the optimization of resource utilization and effective load balancing between the VMs residing in physical servers and among other hosts in the cloud computing environment. But in spite of the growing acceptance of cloud computing, especially due to its scalability and on-demand resource provisioning, poor or inefficient load balancing and resource allocation could lead to suffering from bad performance through underutilization. So, new approaches for load balancing and VM management are necessary. This research presents a novel approach that enhances the efficiency of cloud computing systems by using COA. Focused on the planned and effective hunting techniques of the crocodiles, COA handles the limitations present in traditional load balancing methods and improves resource consumption in cloud environments. The proposed approach optimally distributes tasks across VMs with respect to their processing capabilities and workload. This technique ensures that the dynamic task assignment in the cloud architecture is balanced in such a way that the resource utilization is maximum, but at the same time, response time is minimum. Additionally, the strategy uses VM management techniques which range from migration to scaling up and down based on the variation of workload conditions. The system is able to vary the allocation of the number of resources as per the prevailing demand, hence making it have an optimal utilization of computational resources for the best performance. In this case, the effectiveness of the proposed approach is justified through running simulations with the commonly known cloud computing simulation tool, CloudSim. This paper presents a comparison of the COA-based load balancing and VM management strategy with that of traditional load balancing algorithms in terms of resource consumption, response time, and scalability in order to establish the impact on resource availability through the hypervisor. Results obtained have shown that this technique, based on COA, outperforms all the current existing load-balancing algorithms in resource utilization, response time, and scalability. COA will make the workloads on VMs uniformly distributed so that, for user requests, the response time will be reduced to the maximum to make effective use of cloud resource.

These findings demonstrate the potential of the proposed technique to improve the performance of cloud computing systems

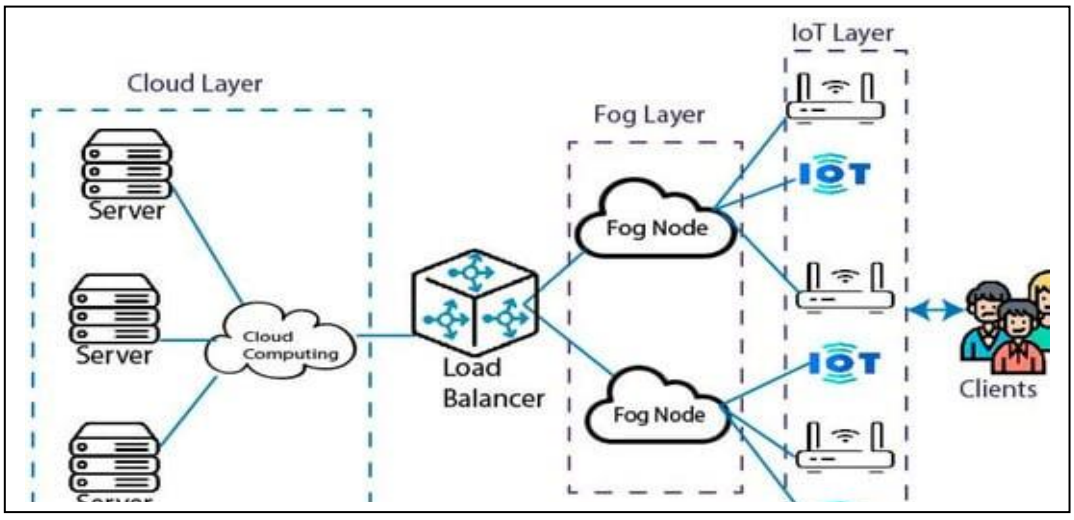

Figure 1. Representation of Load Balancing in Fog Computing [1].

2. Literature Review

Today, cloud computing has matured to be the dominant paradigm of on-demand provision to scalable computer resources to customers globally. Nevertheless, effective utilization of the resources and load balancing remain critical dependencies for the realization of the full potential in cloud environments. In this review, we explored existing literature on virtual machines (VMs) and load-balancing techniques in cloud computing, with proper focus

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

and integration of the Crocodile Optimization Algorithm (COA) for better effectiveness. In another recent research [2], scientists proposed a new approach to load balancing in cloud data centres. They examined dynamic algorithms of workload distribution that could bring the best effect of resource use while at the same time decreasing the reaction time. Soni and Kalra proposed that the differences in variable workloads and resource allocations should be solved with the help of modern algorithms, which improve efficiency and performance. For that, they propose efficient bin-packing-based dynamic load balancing and VM reconfiguration that can be adopted for the whole cloud architecture to optimize resource allocation. Another work that surely guarantees optimal provisioning of resources and system stability in cloud environments is the study of Komarasamy and Muthuswamy [3], where dynamic reallocation of jobs across the VMs with dynamically changing conditions of the workload has been realized. They describe a unique approach to the algorithm for mobile cloud networks load balancing, applying the method of multi-objective optimization [4]. Their work specifically focuses on the problem of the load balancing issue in mobile cloud systems at a time when the limitations of resources have been recognized along with network dynamics as key issues in this area. Cloud computing has brought in a revolutionizing strategy of load balancing through ant colony optimization techniques. The authors suggested the use of ant colony optimization methods for dynamically allocating the job to maximize the cloud system utilization of resources [5]. Leveraging the collective intelligence of an ant colony provides efficient load balancing and further enhances scalability and performance. Some other innovative approaches [6] include the use of the dominant sequence clustering algorithm and mean shift clustering algorithm in job scheduling and load balancing in cloud environments. He introduces in this paper new clustering algorithms that maximize job distribution and resources, while at the same time, ensuring good system performance and scalability. Further, another work [7] has jsones that tries to achieve optimal resource allocation and storage management in cloud environments. This approach provides load balancing, distribution, and storage in an optimized way by use of cloud computing, which enhances the overall efficiency of the system and prompt responsiveness to changing workload conditions. An advanced solution for dynamic load balancing in cloud computing is presented in [8]. The jsons propose adaptive load balancing algorithms that have been developed in order to cope with load variance of workload needs in cloud settings and ensure appropriate resource utilization. They proposed Hermit in response to the load balancing challenges in cloud computing, and another technique is CMODLB, which is a technique that attempts to address the load balancing challenges in the cloud environment [9]. Purpose: optimize the workload and resource allocation in cloud infrastructures so as to improve the performance and scalability of the system.

Work [10] proposes the technique of a meta-heuristic for balancing the load in cloud computing. The authors investigate the use of meta-heuristic optimization techniques to dynamically tune resource allocation and task scheduling for improved efficiency and system responsiveness with varying workload conditions. The research area has attracted due attention, stressing the aspect of load balancing coupled with improving fault tolerance in order to enhance the dependability and efficiency of the service. A new approach for adaptive fault tolerance during load balancing has been proposed in the literature [11]. This approach functions dynamically with changes in demand, which definitely assures good availability and performance, even when there are failures of a system or even when there are unexpected surges of loads. A novel load balancing algorithm for hybrid bee algorithm (HBA) and ant colony optimization (ACO) is introduced in [12]. Available resources can share the workload effectively in the cloud computing environment to yield an improved system performance that has highly utilized all available resources. In this work, the authors in [13] propose using an evolutionary technique, specifically the Genetic Algorithm (GA), to solve the problem of load balancing in cloud computing. Their GA-based strategy strives to maximize resource consumption by imitating natural selection, which results in increased distribution of computing activities with decreased reaction time and energy consumption. The work described in [14] uses a bio-inspired technique for VM load balancing and resource management within the cloud environment. This novel technology is inspired by the working of biological systems that adapt and mimic strategies from natural processes of dynamical equilibration to handle workloads and resources. This style proves the efficiency of bio-inspired algorithms toward the realization of adaptive, robust, and efficient building up of cloud computing infrastructures. Table 1 outlines a comprehensive summary of the most relevant and innovative load-balancing approaches in cloud computing and IoT environments. In the table, the reader can see different approaches toward addressing problems surrounding resource management, efficiency, and optimization of performance that the researchers assumed. Each entry summarizes the main purpose, methodological approach, important contributions, and accomplishments in the linked studies, displaying the evolution and depth of research in that area. In light of these perspectives, the present research assumes profound relevance in underlining the focus on algorithmic innovation and the use of advanced computational tools to deal with the complexity of load balancing. The next research and evaluation studies prove the proposed solutions' effectiveness when benchmarked with existing performance measures and reveal that the proposed solutions have great promise in upscaling durability, efficiency, and optimization capabilities within cloud computing and IoT systems.

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

Table 1: Comprehensive Overview of Various Innovative Approaches to Load Balancing in Cloud Computing

Ref.	Aim and Objective	Contribution	Method/Algorithms	Achievements	Evaluations
1	Develop an optimized framework for load balancing in IoT.	Proposed a dynamic load-balancing framework.	Optimized, dynamic balancing techniques.	Enhanced resource management in IoT.	Compared with existing solutions.
2	Introduce a new load-balancing method for cloud data centres.	Developed a novel approach for load distribution.	A novel algorithm for cloud data centres.	Improved efficiency and performance.	Performance metrics and comparisons.
3	Offer a dynamic load balancing solution with bin packing and VM reconfiguration.	Combined bin packing with VM reconfiguration for efficiency.	Effective Bin Packing and VM Reconfiguration Techniques.	Better resource utilization and balancing.	Efficiency and performance evaluations.
4	Develop a load-balancing algorithm for mobile cloud networks.	Multi-objective optimization for load balancing.	Multi-objective optimization approach.	Improved network performance.	Simulation-based evaluations.
5	Implement load balancing based on ant colony optimization.	Ant colony optimization technique for cloud computing.	Ant Colony Optimization.	Enhanced load balancing in cloud environments.	Comparative analysis with other methods.
6	Develop task scheduling and load balancing using clustering algorithms.	Use of clustering algorithms for improved scheduling and balancing.	Dominant sequence and mean shift clustering.	Enhanced scheduling efficiency.	Algorithmic performance comparison.
7	Propose a novel approach for load balancing in cloud computing.	Load balancing distribution and storage optimization.	Cloud computing-based distribution and storage techniques.	Improved load distribution and storage.	Benchmark comparisons.
8	Introduce "Hermit" for dynamic load balancing in cloud computing.	"Hermit" algorithm for dynamic balancing.	Dynamic load balancing algorithm.	Optimized cloud resource usage.	Performance and efficiency analysis.
9	Offer an efficient load-balancing method in cloud computing.	CMODLB method for efficient cloud load balancing.	Efficient load balancing approach.	Enhanced cloud computing environment performance.	Comparative efficiency evaluation.
10	Develop a novel meta-heuristic approach for cloud load balancing.	Meta-heuristic technique for better load distribution.	Meta-heuristic approach.	Improved load balancing in cloud computing.	Evaluation against standard benchmarks.
11	Create an adaptive fault tolerance method for cloud load balancing.	Adaptive method for fault tolerance during load balancing.	Adaptive fault tolerance techniques.	Enhanced reliability and efficiency.	Reliability and performance assessment.
12	Implement HBA and ACO for cloud load balancing.	Novel algorithm using HBA and ACO.	HBA and ACO techniques.	Improved load balancing in cloud environments.	Algorithmic efficiency and performance metrics.
13	Utilize Genetic Algorithm for cloud load balancing problem.	Genetic Algorithm-based approach for balancing.	Genetic Algorithm techniques.	Optimized load distribution.	Comparative analysis with existing strategies.

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

14	Apply a bio-inspired approach for VM load balancing.	Bio-inspired techniques for efficient resource management.	VM load balancing and resource management.	Enhanced cloud resource management.	Efficiency and performance comparison.
15	Optimize load balancing using a deep learning approach in the cloud.	Integration of deep learning for load balancing decisions.	Deep learning optimization techniques.	Improved decision-making for load distribution.	Efficiency and effectiveness analysis.
16	Model an optimized approach for cloud load balancing.	Optimized model for better resource management.	Modelling techniques for optimization.	Enhanced performance and resource utilization.	Performance metrics and comparative studies.
17	Enhance the selection of load-balancing algorithms dynamically in the cloud.	Dynamic selection mechanism for algorithms.	Dynamic algorithm selection approach.	Increased flexibility and efficiency in load balancing.	Evaluation of selection mechanism effectiveness.
18	Develop an energy-aware load-balancing strategy with VM migration.	Energy efficiency combined with effective load balancing.	Prediction-based VM migration for load balancing.	Reduced energy consumption and balanced load.	Energy efficiency and load distribution evaluation.
19	Use a novel hybrid scheduling algorithm for load balancing in the cloud.	Hybrid scheduling for improved load balancing.	Hybrid scheduling algorithm.	Optimized load balancing and resource allocation.	Comparative analysis with existing algorithms.
20	Propose a new load-balancing technique for VM cloud computing.	New technique for VM load distribution.	Load balancing technique for VM environments.	Enhanced cloud computing performance.	Performance and efficiency assessment.
21	Implement a machine learning-based VM distribution for load balancing.	Machine learning approach for efficient VM distribution.	Machine learning-based dynamic resource mapping.	Improved load balancing and resource utilization.	Effectiveness and efficiency evaluation.

3. Proposed Approach

3.1 COA-Based Load Balancing

In order to enhance the efficiency of cloud computing using the Crocodile Optimization Algorithm (COA), a mathematical framework is necessary. This framework includes key parameters, variables, and equations the following is the abstracted mathematical model for the proposed approach:

Mathematical model parameters:

- N: Total number of virtual machines in the entire cloud architecture.
- M: Total tasks that will be allocated to the VMs.
- P_vm: Processing power of the individual VMs.
- L_vm: Current load on individual VMs.
- Ti: Workload of every task.
- R: Maximum allowed time for response.
- U: Resource utilization threshold.
- S: Scaling factor of VM.
- mig_cost: Cost to migrate a VM.

Variables:

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

- x_ij: Binary variable indicating to which VM every task i is mapped.
- w_j: Workload of the VM j.
- t_ij: Binary variable depicting if the task 'i' is migrated to VM 'j'.
- s_j: Scaling factor for VM 'j'.
- m_ij: Binary variable depicting if the VM 'i' is migrated to VM 'j'.
- Objective function: To minimize the response time while maximizing the resource utilization.

Minimize: ∑(i=1 to M) ∑(j=1 to N) t_ij * R - ∑(j=1 to N) U * (1 - (P_vm * m_ij * w_j))

- Under the constraint:
- Every task should be executed on only one VM: ∑(j=1 to N) x_ij = 1, ∀i∈[1, M].
- For each VM, its workload should not be more than its processing capability: w_j ≤ P_vm * (1 + L_vm), ∀j∈[1, N].
- The following set of constraints ensures that the task 'i' is assigned to VM 'j' if the task migrates to VM 'j': x_ij ≥ t_ij, ∀i∈[1, M].
- Ensure the migration of the task to only one VM: ∑_(j=1)^N t_ij = 1, ∀i∈[1, M].
- Ensure that the VM 'i' has the possibility to be migrated to only one VM: ∑_(j=1)^N m_ij = 1, ∀i∈[1, N].
- Number of VM migrations is limited to a cost with the following constraint:
- The scaling factor should be active for migration to take place.

This should encapsulate the main approach to load balancing and VM management in the cloud computing environment using the Crocodile Optimization Algorithm (COA). It will need a few modifications and fine-tuning with regard to the exact specifics of using COA for implementation and simulation. The adaptation of the Crocodile Optimization Algorithm (COA) using cloud computing is carried out by developing a mechanism of intelligent task allocation to the Virtual Machine (VM) based on the processing capabilities and the current load at the VM. The processes are elucidated in the following steps:

1. Initialization: First, the crocodile population pool will be initialized to represent the possible solutions for the task allocation problems. VM task allocation will be done based on a population of crocodiles. Then, number the maximum iterations of crocodiles and the criterion of convergence.
2. Calculation of Fitness: Evaluate the fitness of each crocodile by applying the objective function defined within the mathematical model, which aims to minimize response time and maximize resource utilization.
3. Movement: Each crocodile, corresponding to a candidate solution of the task allocation problem, evaluates their fitness and decides whether to move from their current position. This aim of movement is to enhance the overall fitness of the crocodile population, equating to different task allocations in the context of load balancing, where movement involves changing the assignments of tasks to VMs.

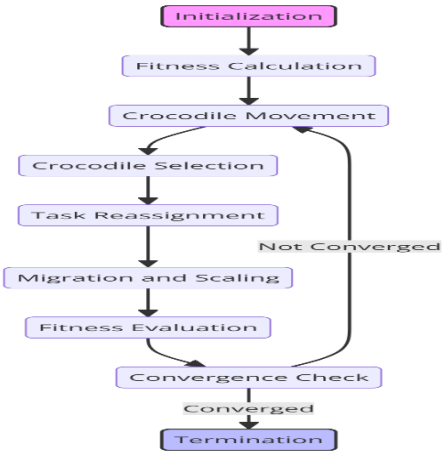

Figure 2. Proposed Crocodile Optimization Algorithm (COA) For Load Balancing in Cloud Computing

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

4. Crocodile Selection: Choose a subset of crocodiles for potential movement based on their fitness levels. Crocodiles with higher fitness are more likely to be selected for movement. This selection process mirrors the identification of task allocation solutions that demonstrate potential in reducing response time and enhancing resource utilization.

5. Task Reallocation: Those crocodiles that are selected and supposed to be moved from one location to another go through a task reallocation phase. Task reallocation refers to the realigning of tasks that have been previously assigned from current VMs to new VMs. Decisions regarding task reallocation are driven by the optimization mechanisms of the COA.

6. Migration and Scaling: In cloud computing, the migration of tasks is referred to as the transfer of any task to another VM. The current workload and processing capabilities on the VMs are taken into consideration by the migration process. The COA can employ migration strategies inspired by the behavior of crocodile hunting for the optimal allocation of tasks. It will also include VM scaling in the migration strategy if agreed. VM scaling takes into consideration the reshaping of the capacity for VMs based on the prevailing workload.

7. Fitness Evaluation: After reallocation (migration), the fitness evaluation of the migrated crocodiles with the new allocated tasks will need to be recalculated. It measures the change impact with regard to response time and resource utilization.

8. Convergence Check: It must be checked whether the algorithm has converged or the maximum number of iterations is reached. The convergence means that solutions of task allocation have become stabilized, and further optimization is not likely to provide observable gains.

9. Termination: If the algorithm has not yet converged, the process returns to step 3 for further iterations. Conversely, if convergence has been achieved or the maximum iterations have been reached, the algorithm is terminated. The final task allocation solution(s) obtained will represent an optimized load-balancing configuration.

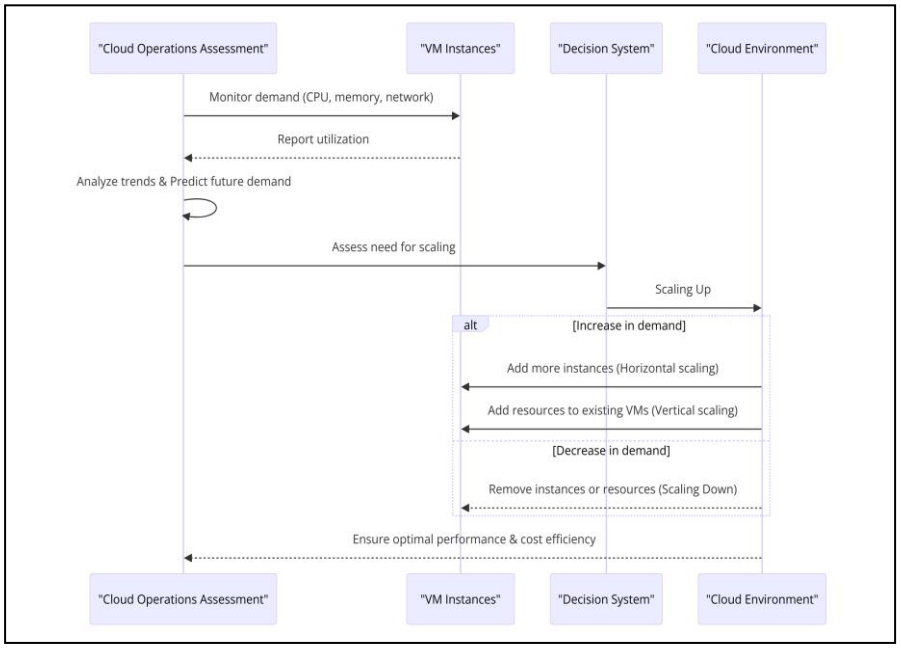

Figure 3. COA for Predict Future Demand

3.2 VM Management with COA

Demand and Utilization Assessment:

COA keeps track of the current VMs demand; this is done by continuously monitoring and analysing metrics such as CPU load, memory usage, and network traffic. Real-time monitoring provides COA with dynamic reactions to any change within the workload. It can analyse the trend in the usage of resources and thus predicts future demands using the Crocodile Optimization Algorithm (COA). This predictive capacity is based on historical data analysis that gives conscious scaling decisions (Figure 3).

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

Decision Making for Scaling:

Scaling up: COA will then assess whether there is a need for scaling up, but in any event, the projection of the increase in demand from COA should be tested against its effect on the current resources. This may be through instances of virtual machines (horizontal scaling) or through adding more resources to the already available virtual machines (vertical scaling), depending on the kind of capability and requirement imposed by the cloud environment. Scaling down: On the other hand, in cases where the demand for their services is low, COA can initiate scaling down to minimize its cost and reduce its energy consumption. This decision is very measured in order to guarantee that performance is not jeopardized and the best balance between efficiency and service quality is reached. Achieving Optimal Performance and Cost Efficiency: Integration of COA with Scaling Strategies of Virtual Machines Integrated with the COA, the Virtual Machine Scaling Strategy empowers a cloud environment to easily conform to fluctuating demands. It is supposed not only to maintain high performance and service availability but also to optimize operational expenses by avoiding resource over-provisioning and under-utilization. Cloud environments will be able to reach high levels of performance and cost efficiency given the proper use of the resource provided to them.

4. Result and Discussion

In this section, we contrast the proposed Crocodile Optimization Algorithm (COA) with two existing approaches, HMLB and QMPSO-LB, based on several performance metrics, such as latency, bandwidth utilization, and completion time, throughput, energy, CPU, and memory uses, overhead time, and SLA violations. We consider two cases following the task- and virtual machine (VM)-based approach of the COA model. The main reason for the COA model to reduce latency had been attributed to the use of a bi-class neural network. This comes in the way of a network where tasks are categorized, hence facilitating ease in scheduling and providing speed and efficiency. Thus, COA attains speedier task scheduling outcomes due to the superior convergence. Such benefits of COA are explored in Table 2, with a detailed comparison of latency in respect to different numbers of tasks. In this paper, performance metrics and benefits of COA will, therefore, be compared with the existing models, HMLB, and QMPSO-LB, to furnish detailed performances and advantages of COA with respect to task scheduling.

Table 2: Numerical Analysis of Latency for a Comparative

Latency	COA	**HMLB[23]**	**QMPSO-LB[24]**
5	16	25	10
10	38	50	25
15	61	85	38
20	92	120	50

Three repositories are employed by the VM monitor to help in effective load balancing. First is a primary repository meant to store search history for virtual machines (VMs). The hierarchical structure of the repository is designed to store the minimum storage loads and guarantee good performance. Second is the use of secondary repositories in VM indexing to help in the selection of appropriate VMs for operations of load balancing? Third is the realization of a recovery repository to duplicate data from primary and secondary repositories. The search time and latency are reduced due to the elimination of the duplication; hence overall significant efficiency improvement is realized. Analysis showed that COA model demonstrated better improvement in bandwidth utilization compared to the prevailing model, HMLB, and the QMPSO-LB. The other focus that the discussion takes on is a comparative analysis of the effectiveness of COA in bandwidth optimization against its counterparts. The three repositories that exist in the COA model are very instrumental in the realization of these efficiencies that are much needed in bandwidth usage.

Table 3: Numerical Analysis of Bandwidth Consumption

Latency	COA	**HMLB**	**QMPSO-LB**
5	25	18	30
10	30	32	60
15	51	57	90
20	60	73	140

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

Tables 2 and 3 present the comparison of the consumed bandwidth and latency for three models: COA, HMLB, and QMPSO-LB. This shows that COA is better in every case than QMPSO-LB and more efficient than HMLB, especially for fewer task counts. These results demonstrate COA's superior management of bandwidth and its ability to achieve lower latency. In addition, from the trend, COA keeps lower latency at an increased number of tasks, hence proving it scales well and is efficient even at bigger workloads compared to HMLB; hence, QMPSO-LB. The proposed COA model significantly reduces the time required to complete tasks, way ahead of the existing models, as reflected in Table 4. COA makes this possible through the classification of tasks to make the processes easier, and, with that, it proceeds to use a fast convergence algorithm to schedule efficiently. In addition, the three repositories in the load balancing implementation reduce the completion time based on optimal task allocation and reduction of search latency. As such, COA performs better, especially when the analysis is conducted based on computational times that are much faster than HMLB and QMPSO-LB. These could be attributed to the innovative strategies in task classification, scheduling, and resource management employed by the COA model.

Table 4: Numerical Analysis of Completion Time

Latency	COA	**HMLB**	**QMPSO-LB**
5	3	3	5
10	9	15	18
15	13	24	29
20	21	35	40

Enhancing the Quality of Service (QoS), throughput measures the rate of successful task transmissions over time. The COA model outperforms compared to traditional models in minimizing throughput time, as clearly shown in Table 5, in dealing with high-throughput operations. It operates efficiently through different mechanisms, including bi-class task classification, QoS-focused scheduling, multi-repository-based load balancing, and congestion-aware task allocation. The bipartite graph, in VM allocation, optimizes matching processes, which again reduces the further latency of allocation and throughput time. This provides COA with an advantage over scenario-specific comparisons; that is, COA outperforms comparisons of the HMLB and QMPSO-LB models. From Table 4, the COA model has the best performance in reducing the throughput time, compared to the HMLB and QMPSO-LB models, with respect to the different volumes of tasks. The COA model constantly held lower latencies with an increasing number of tasks, which is a good show that COA is effective for quick task processing. In fact, this is a show of marked efficiency, especially in those scenarios with huge volumes of tasks, against all other models but COA. These results underscore the effectiveness of COA in managing high workloads and enhancing system throughput.

Table 5: Numerical Analysis of Throughput Time

Latency	COA	**HMLB**	**QMPSO-LB**
5	12	20	36
10	28	53	62
15	48	74	84
20	63	91	110

This is the measure of energy consumption by devices in the completion of tasks. The COA model shows low energy consumption compared to existing models. This has been made possible through the pre-scheduling task classification approach taken by the COA model. By grouping tasks with similar characteristics, scheduling delays are minimized, resulting in reduced energy usage. The energy efficiency of the COA model is further delineated in Table 6, for it shows that the COA model seems to consume significantly less energy than the HMLB model by 26KJ and the QMPSO-LB model by 51KJ. These findings demonstrate that COA has been proven effective in optimizing the energy consumption of the tasks and their schedules.

Table 6: Numerical Analysis of Energy Consumption

Latency	COA	**HMLB [23]**	**QMPSO-LB[24]**
5	12	10	24
10	35	54	68

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

15	99	120	136
20	152	173	198

One of the principal metrics that actually measures efficiency in using resources during all processes: higher usage indicates less wastage and better efficiency of the whole system. The proposed COA model optimizes CPU and memory usage through the mechanisms of task classification, scheduling, load balancing, and allocation. It basically works towards less wasting of resources and better efficiency of the system. In other words, the existing models are inefficient: their job scheduling and assignment of resources can be achieved with lower efficiency, i.e., consuming more resources, for a suboptimal latency that is increased. Table 7: CPU and Memory Utilization vs. Task Counts. It shows the dominance of the COA model in efficient resource utilization. It can be observed from the results that COA is more efficient than existing models in CPU and memory management, resulting in a better system performance output with less waste of resource.

Table 7: Numerical Analysis of CPU & Memory Utilization

Latency	COA	**HMLB**	**QMPSO-LB**
5	5	3	1
10	14	10	2
15	20	18	7
20	27	25	10

5. Conclusion

We propose an idea in this research with the intention of improving the performance of cloud computing systems through load balancing and VM management using the Crocodile optimization algorithm. This paper is going to focus on how efficient resource utilization and effective load balancing can help in realizing optimal performance in cloud computing. The COA algorithm takes its inspiration from crocodile hunting behavior and provides, dynamically, a strategy in terms of task allocation and VM management. The COA algorithm offers relatively good performance when compared with other classical algorithms on simulation, hence the sense of practicality is very good in achieving better performance in the cloud computing system. In summary, the paper is of great significance for effectiveness in cloud computing through the application of the Crocodile Optimization Algorithm (COA) in load balancing and VM management. Inspired by the hunting behavior of crocodiles, COA innovatively proposes the approach of dynamic task allocation to VMs according to its capability and its current load for effective load balance and to achieve optimum utilization of resources. This is more so when proven performance of COA in simulations, resource utilization, response time, and scalability over traditional algorithms is a true indicator that indeed it can be used to improve cloud computing systems. This paper makes a compelling case for the broader application of COA in the field of cloud computing.

References

[1] Shuaib, M., Bhatia, S., Alam, S., Masih, R. K., Alqahtani, N., Basheer, S., & Alam, M. S. (2023). An Optimized, Dynamic, and Efficient Load-Balancing Framework for Resource Management in the Internet of Things (IoT) Environment. *Electronics, 12*(5), 1104. https://doi.org/10.3390/electronics12051104

[2] Soni, G., & Kalra, M. (2014, February). A novel approach for load balancing in cloud data centers. In *2014 IEEE International Advance Computing Conference (IACC)* (pp. 807-812). IEEE.

[3] Komarasamy, D., & Muthuswamy, V. (2016). A novel approach for Dynamic Load Balancing with effective Bin Packing and VM Reconfiguration in the cloud. *Indian Journal of Science and Technology*, *9*(11), 1-6.

[4] Arun, E., Reji, A., Mohammed Shameem, P., & Shaji, R. S. (2017). A novel algorithm for load balancing in mobile cloud networks: Multi-objective optimization approach. *Wireless Personal Communications*, *97*, 3125-3140.

[5] Alyouzbaki, Y. A. G., & Al-Rawi, M. F. (2021). Novel load balancing approach based on ant colony optimization technique in cloud computing. *Bulletin of Electrical Engineering and Informatics*, *10*(4), 2320-2326.

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024

[6] Al-Rahayfeh, A., Atiewi, S., Abuhussein, A., & Almiani, M. (2019). A novel approach to task scheduling and load balancing using the dominant sequence clustering and mean shift clustering algorithms. *Future Internet*, *11*(5), 109.

[7] Sharma, A., & Sharma, K. K. (2023). A Novel Approach for Load Balancing distribution and storage by using Cloud Computing. In *E3S Web of Conferences* (Vol. 399, p. 04009). EDP Sciences.

[8] Mohapatra, S., Mohanty, S., Hota, A., Patra, P. K., & Dash, J. (2021). Hermit: A Novel Approach for Dynamic Load Balancing in Cloud Computing. In *Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 1* (pp. 275-287). Springer Singapore.

[9] Negi, S., Rauthan, M. M. S., Vaisla, K. S., & Panwar, N. (2021). CMODLB: an efficient load balancing approach in cloud computing environment. *The Journal of Supercomputing*, *77*, 8787-8839.

[10] Mohanty, S., Patra, P. K., Ray, M., & Mohapatra, S. (2021). A novel meta-heuristic approach for load balancing in cloud computing. In *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing* (pp. 504-526). IGI Global.

[11] Tamilvizhi, T., & Parvathavarthini, B. (2019). A novel method for adaptive fault tolerance during load balancing in cloud computing. *Cluster Computing*, *22*, 10425-10438.

[12] Mousavi, S. M., & Gábor, F. (2016). A novel algorithm for Load Balancing using HBA and ACO in Cloud Computing environment. *International Journal of computer science and information security*, *14*(6), 48.

[13] Dalal, S., & Kukreja, S. (2016). Genetic Algorithm based Novel approach for Load Balancing problem in Cloud environment. *International Journal of Computer Science and Information Security (IJCSIS)*, *14*(7).

[14] Assudani, P. J., & Balakrishnan, P. (2022). A novel bio-inspired approach for VM load balancing and efficient resource management in cloud. *International Journal of Ad Hoc and Ubiquitous Computing*, *40*(1-3), 214-224.

[15] Kaur, A., Kaur, B., Singh, P., Devgan, M. S., & Toor, H. K. (2020). Load balancing optimization based on deep learning approach in cloud environment. *International Journal of Information Technology and Computer Science*, *12*(3), 8-18.

[16] 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.

[17] Khara, S., & Thakkar, U. (2017, July). A novel approach for enhancing selection of load balancing algorithms dynamically in cloud computing. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)* (pp. 44-48). IEEE.

[18] Patel, D., Gupta, R. K., & Pateriya, R. K. (2019). Energy-aware prediction-based load balancing approach with VM migration for the cloud environment. *Data, Engineering and Applications: Volume 2*, 59-74.

[19] Domanal, S. G., & Reddy, G. R. M. (2015, November). Load balancing in cloud environment using a novel hybrid scheduling algorithm. In *2015 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)* (pp. 37-42). IEEE.

[20] Chaudhary, D., & Chhillar, R. S. (2013). A new load balancing technique for virtual machine cloud computing environment. International Journal of Computer Applications, 69(23).

[21] Lilhore, U. K., Simaiya, S., Guleria, K., & Prasad, D. (2020). An efficient load balancing method by using machine learning-based VM distribution and dynamic resource mapping. Journal of Computational and Theoretical Nanoscience, 17(6), 2545-2551.

[22] Kareem, S. W. (2022). A nature-inspired metaheuristic optimization algorithm based on crocodiles hunting search (CHS). International Journal of Swarm Intelligence Research (IJSIR), 13(1), 1-23.

[23] Khaleel, M.I., Safran, M., Alfarhood, S., & Zhu, M. (2023). A Hybrid Many-Objective Optimization Algorithm for Job Scheduling in Cloud Computing Based on Merge-and-Split Theory. Mathematics, 11, 3563.

[24] Nebagiri, M. H. ., & Hnumanthappa, L. P. . (2023). Multi-Objective of Load Balancing in Cloud Computing using Cuckoo Search Optimization based Simulation Annealing. International Journal of Intelligent Systems and Applications in Engineering, 12(9s), 466–474.

DOI: https://doi.org/10.54216/FPA.170205
Received: January 20, 2024 Revised: April 12, 2024 Accepted: September 17, 2024