Improving Task Scheduling in Cloud Datacenters by Implementation of an Intelligent Scheduling Algorithm

Omer K. Jasim Mohammad*, Bassim M. Salih
University of Fallujah, Fallujah Aar, Iraq
E-mail: Omerk.jasim@uofallujah.edu.iq, Bassim.ms1977@uofallujah.edu.iq
*Corresponding author

Keywords: cloud computing, cuckoo intelligent algorithm, task scheduling, intelligent scheduling algorithm, cloud datacenter

Received: March 5, 2024

The need for mobile and online applications and services has resulted in a significant expansion of cloud computing services. The exponential expansion emphasizes the significance of minimizing scheduling time and optimizing resource utilization in a dynamic environment. Therefore, several scheduling algorithms have been developed to tackle these issues by utilizing intelligent scheduling methods, such as Genetic Algorithms, greedy algorithm, Antlion Optimizer, Ant Colony optimization, and Cuckoo Intelligent Algorithm. This paper presents a comprehensive analysis of intelligent optimization methodologies, with a particular emphasis on the Cuckoo intelligent methodology. Furthermore, it introduces a suggested deployment of a Cuckoo-based cloud computing system as a highly effective algorithm that is expected to produce enhanced outcomes in work scheduling.

Povzetek: Članek predstavlja inteligentni algoritem za razporejanje nalog v oblačnih podatkovnih centrih, ki temelji na Cuckoo inteligentni metodologiji. Avtorja podrobno analizirata različne optimizacijske metode, kot so genetski algoritmi, požrešni algoritmi, Antlion optimizator in optimizacija kolonij mravelj. Predlagana uporaba algoritma, temelječega na Cuckoo metodi, naj bi izboljšala čas razporejanja in optimizacijo virov v dinamičnih okoljih, kar bi prispevalo k večji učinkovitosti oblačnih storitev.

1 Introduction

Cloud computing has arisen as a solution to the demand for outsourcing computer and storage resources to customers. The quick expansion of this phenomenon can be attributed to the significant advancements in communication technologies and virtualization technologies. As a result, many of its systems have become indispensable in our daily activities [1]. Furthermore, it provides clients with reusable, efficient, and distributed resources in three various service forms: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Companies such as Amazon, Microsoft, IBM, and Google offer cloud services, allowing non-experts to readily access the resources they require without needing to understand the complexities of cloud infrastructure [2, 3]. The task scheduling process is critical to cloud computing's continuously changing state, significantly affecting performance and resource use. The need for cloud services and apps is growing.

Thus, it's vital to use clever algorithms for work scheduling. Consequently, the Cuckoo algorithm is used in this work to offer a task-scheduling enhancement technique [4, 5]. This research also laid a considerable stress on determining the most effective method involves to schedule jobs with varying workloads, Different reservoirs, and unanticipated tasks in light of the dynamic state of cloud computing. The suggested enhancement dynamically and effectively distributes work around the cloud data center using the Cuckoo intelligence algorithm. It fundamentally revolves around merging artificial intelligence with optimization techniques for enhancing outcomes. This will optimize the consumption of resources, minimize latency, and improve the overall performance of the system [6, 7].

This research holds great importance not only in terms of operating efficiency, but also in meeting user expectations, assuring cost-effectiveness, and inspiring sustainability by increasing energy usage. The research analyzes the theoretical basis of intelligent algorithms, their suitability for cloud work scheduling, and provides actual evidence to substantiate the effectiveness of the proposed improvement. This research adds to the ongoing discussion in cloud computing by providing insights into novel methods of job scheduling that have the potential to significantly transform cloud datacenter operations.

The rest of paper is orchestrated as follows: Section II showed the study borders. Section III reviewed of the literature survey. In Section IV the details of the Cuckoo algorithm: definition and algorithm are explained, Section V shows the main building of proposed work. The experimental environment of cloud and outcomes have been shown in Section VI, and Section VII discussed the obtained results. Section VIII showed the research gaps and limitations. Finally, Section IX illustrated the conclusion.

2 Study borders

2.1 Objectives

The objectives of the proposed study are;

1. To enhance cloud computing services reliability and performance.
2. In order to improve resource allocation and lessen virtual machine overloading cloud computing environments should optimize job scheduling.
3. To research and assess dissimilar work scheduling algorithms, such as metaheuristic techniques (e.g., Genetic Algorithms, Particle Swarm Optimization), heuristic techniques (e.g., Round Robin, min-min), and classical methods (e.g., First-Come-First-Serve).
4. To suggest and assess a method for work scheduling in cloud datacenters based on the cuckoo search algorithm.
5. To evaluate how well cuckoo search algorithms function at increasing the efficiency of task scheduling and the ability to adjust to changing and unpredictable workloads.
6. To show off the advantages of employing cuckoo search algorithms for effective resource management in cloud computing, especially with regard to virtual machine utilization.

2.2 Problem statement

The issue is the necessity to further develop distributed computing administrations' presentation and unwavering quality, which are turning out to be progressively fundamental for contemporary web-based applications. Improving position planning for distributed computing settings is basic to achieving this point since it upgrades asset designation and diminishes virtual machine overpopulation. Finding the best technique is troublesome, however, in light of the fact that there is an extensive variety of undertaking planning calculations open, from cutting edge metaheuristic and heuristic systems to traditional ones. By looking at and surveying the cuckoo scan calculation's pertinence for work planning for cloud server farms, the recommended research tries to address this trouble in this specific circumstance. The principal objective is to assess the way that well cuckoo search calculations might work on the adaptability and productivity of occupation booking, in this way supporting upgrading distributed computing asset usage and administration quality because of dynamic and unforeseen responsibilities.

2.3 Study scope

The study's scope contains an in-depth examination of how to improve cloud computing services' dependability and performance through job arrangement that is optimized. In order to mitigate virtual machine overpopulation in cloud computing settings and handle resource allocation difficulties, the project intends to investigate and assess several work scheduling techniques. This covers a detailed examination of empirical strategies like min-min and Round Robin, Meta heuristic strategies like Genetic Algorithms and Particle Swarm Optimization, and traditional approaches like First-Come, First-Served [8, 9]. The cuckoo search algorithm provides the groundwork for developing an original approach for task scheduling in cloud datacenters. The new method for task scheduling in cloud datacenters is modelled and their performance is estimated. Such methodic preference is compelling due to the fact that it ensures the task scheduling that is robust and flexible in environments with the dynamic loads. The purpose of the research is to provide basic knowledge of cuckoo search algorithms that help in scheduling performance and adaptability by evaluating compatibility with efficiency and adaptive properties. The study will take cloud computing simulated conditions for research by performing experimental analyses of the algorithm proposed for the cuckoo search strategy. The process will take form of comparing performance of current scheduling strategies with our scheduling method. Also, there will be evaluation of the weak and strong points of the strategies and comparisons will be made between them and with the current criteria of the classes' schedules. This study shows how suitable is the Cuckoo search algorithm as well at different workload intensities adopting to restricted resources. In addition, the scope of research encompasses determining the broader impacts of use of cuckoo search algorithms for job scheduling cloud data centers. Such review encompasses an assessment of their impact on cost-efficiency, resource allocation, and the IT infrastructure including reliability [10, 11]. Amongst the main purpose of the research is to help with understanding the various factors involved in job scheduling procedures and switch to more optimized approaches for the cloud computing environments. The study intends to achieve both a knowledge enriching and a practical application of the work scheduling techniques with a theoretical foundation and a screening by realistic testing. The research that puts a great deal of emphasis on performance improvement, reliability enhancement and resource efficiency with a view to complementing implementation of the cloud computing services thorough a consecutive and systematical analysis.

3 Review of the literature

In the cloud computing, load balancing across dispersed systems and resource optimization depends heavily on effective job scheduling, for example,

Moradbeiky and Bardsiri [12] provides supplementary way to address task planning for cloud conditions harnessing the Cuckoo Development Calculation. The aim of the review was to lessen work finish time and augment asset portion by advancing undertaking planning effectiveness in distributed computing. The review's objectives were to figure out how well the Cuckoo Streamlining Calculation functions for task booking, contrast its exhibition with additional customary methodologies, and measure how well it can follow conditions with a variable responsibility. The measurements that were harnessed to measure the factors were calculation combination speed, asset usage, and assignment achievement time. As per the review's discoveries, the Cuckoo Improvement Calculation outflanked traditional procedures as far as asset exploitation and undertaking fruition time, showing empowering brings about further developing errand planning proficiency. Furthermore, its adaptability in obliging changing responsibilities was referred to as a key advantage, showing its true capacity for valuable application in distributed computing settings.

Vijindra and Shenai's [13] focused on planning issues in distributed computing. The essential points of this exploration are to look at and assess the different planning issues that emerge in distributed computing settings. The's authors will likely pinpoint the basic components —, for example, load adjusting, task prioritization, and asset allotment that influence booking effectiveness. The paper evaluates different planning techniques and calculations utilized in distributed computing frameworks by directing a careful survey of the group of current writing and contextual analyses. The manner in which these factors influence execution estimations like throughput, response time, and asset usage is the way they are estimated. The review's decision underscores the meaning of proficient booking strategies for augmenting asset use, helping framework viability, and raising distributed computing conditions' overall effectiveness. The authors likewise stress the need of extra review and headway in this field to deal with new issues and consider changing distributed computing ideal models.

Abadi, Z. J. K., and Mansouri, N. [14] completed a broad examination on fluffy framework booking procedures in disseminated conditions. The reason for the review was to explore and evaluate fluffy based booking calculations' presentation in appropriated processing conditions. The aims were to distinguish patterns and issues in the field, break down various booking calculations utilizing fluffy frameworks, and survey the assortment of existing writing. Computational intricacy, calculation execution measurements, and climate versatility were among the viewpoints that the authors evaluated. The review arrived at the resolution that fluffy based planning calculations, which give adaptability, heartiness, and flexibility to changing responsibility conditions, offer expected choices for advancing assignment planning for scattered settings.

Moreover, task scheduling algorithms for energy optimization in cloud environment: a comprehensive review was done by Ghafari, R., et al. [15]. The objective of the review was to completely look at present place of employment planning techniques comparable to distributed computing energy enhancement. The aims were to characterize and recognize different work planning calculations, survey how well they limited energy use, and pinpoint potential regions for extra review and improvement. Algorithmic execution pointers like energy utilization, task fruition time, and asset use effectiveness were among the factors surveyed. The examination reached the decision that despite the fact that various work booking calculations have been invested on a mission to streamline effort use in cloud frameworks, more careful investigations and correlations are as yet expected to figure out which techniques are the best. Moreover, the review exhibited that it is so essential to take into for making energy-proficient undertaking booking calculations for distributed computing, various perspectives, including responsibility attributes, asset accessibility, and framework heterogeneity, ought to be thought about.

The concentrate by Murad, S. A. et al. [16] looked at distributed computing position planning strategies and recommends an shrewd structure in light of need rules. The paper expects to analyze present place of employment booking techniques, identify their advantages and disadvantages, and furnish another structure with astute prioritization standards for compelling undertaking planning for cloud conditions. The presentation measurements of work booking calculations, for example, make length, asset use, and response time, are among the factors that are assessed. The review arrives at the resolution that smart prioritization rules can be integrated into work booking structures to significantly build the viability of asset portion and by and large framework execution in distributed computing settings. This would at last further develop client nature of administration.

The presentation examination of occupation planning calculations in distributed computing frameworks is the primary subject of Walia and Kaur's study [17]. The principal objectives of the review are to decide what different undertaking booking calculations mean for framework execution boundaries including execution time, asset use, and by and large throughput, as well as to evaluate how compelling each approach is. Framework throughput, memory use, computer chip use, and calculation execution time are among the measurements that are estimated. The review shows that different errand planning calculations display fluctuating levels of productivity and adequacy in advancing asset portion and limiting framework overheads in distributed computing conditions through exploratory assessment and relative examination.

Ajmal, M. S. et al. [18] reviewed the task scheduling objectives to recommend a half and half strategy for cloud server farm work planning improvement that mixes hereditary and insect province improvement methods. The review estimates three factors: in general framework effectiveness, asset usage, and undertaking fruition time. That's what the review's discoveries show, in contrast with customary planning strategies, the half breed insect inherited calculation gives better execution and adaptability, which further develops asset assignment and abbreviates task fruition times in cloud server farm settings [22].

Kashikolaei et al. [19] distributed a review named " An enhancement of task scheduling in cloud computing based on imperialist competitive algorithm and firefly algorithm." The review's objectives incorporate mixing the firefly and radical cutthroat calculations to propose and survey a superior technique for work planning for distributed computing. The objective of the examination is to further develop framework execution in cloud conditions by expanding position planning proficiency and asset usage. Estimated factors include an agenda throughput, asset usage, and undertaking execution time. That's what the review's decision shows, when contrasted with traditional strategies, the crossover approach that utilizes both the firefly and radical serious calculations performs better at task planning and displays more prominent versatility, flexibility, and heartiness while taking care of jobs connected with distributed computing. Table 1 summarizes the main parameters of this section such as algorithm contribution, cloud platform, metric, features, and weakness.

Table 1: Existing studies consider the main scheduling parameters.

Authors /Ref	Algorithm Contribution	Cloud Platform Engine used	Metrics	Features	Weaknesses
Moradbeiky and Bardsiri [12]	Cuckoo Optimization Algorithm	Cloudsim	Length of task, tasks contents, availability (RAM, CPU)	Improved the performance parameters.	The results will not be satisfying enough due to neglecting the resource arrangement.
Vijindra and Shenai's [13]	ALO, Swarm	None	Execution time, load balancing	Draw a vision of the main parameters for scheduling algorithms.	This work is purely a review and does not involve the use of any actual cloud platform. The findings are based only on survey work.
Abadi, Z. J. K., and Mansouri, N. [14]	Fuzzy-based scheduling algorithms	Open eucalyptus cloud computing cloud environment	Comparing the fuzzy-based scheduling schemes based on merits and demerits,	Enumerated many fields in which fuzzy logic is employed in distributed systems.	It is necessary to compare all the results gained with other algorithms that employed the prediction scheme in their research.
Ghafari, R., et al. [15]	Comparative study of 67 scheduling	Amazon Cloud Infrastructure	Quality of Service (QoS) factors (e.g., execution time and cost)	Categorize the algorithms into heuristic-based task scheduling, meta-heuristic-based task scheduling, and other task scheduling techniques.	The results will be unsatisfactory as a result of disregarding resource allocation and inefficient algorithm integration.
Murad, S. A. et al. [16]	Job Scheduling techniques (JST)	Real environment	response time, makespan time, flow time, finish time, cost, and resource utilization	Enhance IoT connectivity by creating a streamlined and sophisticated protocol.	The results indicate an uneven distribution of the burden.
Walia and Kaur's study [17]	Comparative study among: GA, FPA, ACO, and TS-GA	Cloud-ASP.NET	Resource Utilization, Energy Consumption	Choose the most suitable meta-heuristic scheduling algorithm based on resource utilization, completion time, energy consumption, and computing costs.	The system lacks comprehensive results for all scheduling scenarios in the cloud environment.
Ajmal, M. S. et al. [18]	Hybrid ant genetic algorithm (HAGA)	Cloudsim	throughput, response time, energy consumption and resources utilization	The system incorporates characteristics from both the genetic algorithm and the ant colony algorithm, while also partitioning jobs and virtual computers into smaller clusters.	The acquired findings are not indicative of the true performance metric since they fail to take into account the allocation of resources.
Kashikolaei et al. [19]	Combination of ICA and FA	Cloudsim	Load balancing, makespan, CPU time, Stability	Enhance the process of scheduling and load balancing by utilizing local and global search methods.	Further optimization is required, particularly in the areas of resource allocation, task scheduling, and workload computation.

4 Cuckoo algorithm

The Cuckoo algorithm (CA) is an intelligent optimization algorithm depends on inspiration from the breeding behavior of cuckoo birds and it's distinguished by uncomplicated nature and effectiveness. The Cuckoo algorithm (CA) is an intelligent optimization algorithm that depends on inspiration from the breeding behavior of cuckoo birds and is distinguished by its uncomplicated nature and effectiveness.

The core premise of CA is based on the reproductive strategy of cuckoo birds, through laying their eggs in the nests of other host birds. This behavior is replicated algorithmically through iterative searching for optimal solutions. The optimal solution summarizes by generating a new solution (referred to as cuckoos) and replacing inferior solutions within the population (referred to as the nest) [20, 21]. The primary essential elements of CA are Eggs and abandonment. Eggs represent the potential solutions, whereas abandonment simulates the rejection of less favorable ideas. Furthermore, the operational process of CA can be summarized as a combination of levy flight and nest selection. CA frequently utilizes Levy flights to simulate the stochastic movements made by cuckoos, enabling rapid exploration of the solution space and avoiding getting trapped in local optima. The nest selection process involves determining and evaluating the best solutions (fitness) depending on the objective function of the optimization problem. Mathematically, in order to understanding the mathematical model of CUCKO Algorithm see [22, 23]. Algorithmically, the pseudo code of CA listed as follows:

Initialize the cuckoo with some random number of egg (n),
Calculate the set of egg.
Find the best fit egg location and size.
While (the end criterion is not satisfied)
Update Laying eggs in host birds, determined destroy eggs, growing egg
For every egg
Select habitat of each newly cuckoo.
Remove birds' nests wrong.
Create cluster population
Travel to best cluster
Calculate optimal fitness.
End for
Replace an egg position with its corresponding egg position it if becomes fitter
Update egg location if an egg becomes fitter than the selected eggs-location.
End while
Return optimal location

5 Proposed work

Cloud computing optimizes the allocation of resources, mitigates VM overcrowding, and enhances the quality of Cloud computing has provided a wide range of internet services such as storage, software, infrastructure, and software development. Various software apps, like Google Apps and Microsoft Office 365, are commonly utilized by regular consumers on a daily basis. Therefore, it is imperative to consistently improve the performance and dependability of cloud computing. Task scheduling is a crucial aspect of performance. Task scheduling in services for consumers. Authors have investigated several algorithms for Task scheduling, ranging from traditional methods like First-Come-First-Serve (FCFS) to heuristic approaches such as Round Robin (RR) and min-min, as well as metaheuristic methods like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) [25, 26]. According to this study, at first, the population was made up of various arrangements of virtual machines. Two separate modes were surveyed in relation to the system. The initial state of the population consisted of various configurations of the machines, whereas the subsequent state consisted of different configurations of the tasks entering the system. In the following stage, the objective function value is obtained for each order, and then the bird starts generating new solutions by randomly assigning eggs to each cuckoo. The solutions involve different configurations of virtual machines that are deployed to perform the task. The aim of this improvement is to utilize an alternative virtual machine on the current host to carry out the task. During the subsequent stage, the optimal settings are revised. The optimal values obtained from each order are partitioned into groups using K-means clustering. Each cluster represents the range of responses collected for each order. The destination is chosen based on clusters with superior rankings, as this area has produced more favorable outcomes compared to other areas. Additionally, a larger number of cuckoos have thrived in this environment. Consequently, populations from other clusters migrate towards this improved location, guided by the assumed probability. This variation in movement prevents the occurrence of falling into a local ideal solution and facilitates the execution of both general search and local search, as well as convergence towards the optimal response. Therefore, the subsequent requests for virtual machines in this domain is generated using the Levy distribution, and the values are once again modified. This tendency continues until it reaches the optimal solution. Cloud datacenter task scheduling benefits from using CA due to its ability to effectively manage dynamic and unpredictable workloads. Through utilizing its innate capacity to explore many solution spaces and avoid suboptimal outcomes, CA can adjust to evolving situations within a cloud environment. CA has the ability to assign tasks to resources that are currently available, taking into account the current conditions in real-time. The ability to adapt is essential for fulfilling diverse demands and optimizing resource efficiency. Thus, as illustrated in figure 1, the utilization of CA during the task scheduling phase enables efficient management of the scheduling problem by using virtual machines (VMs) in a cloud context.

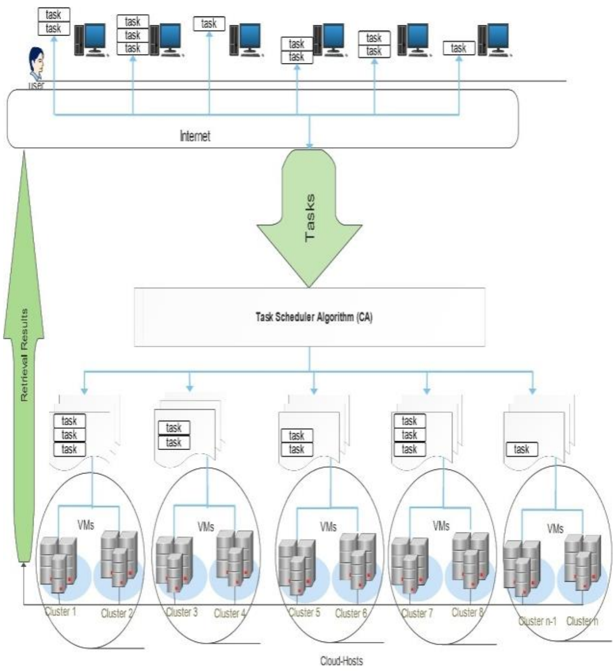

Figure 1: System model

As depicted in Figure 1, each cloud user transmits tasks via the internet using any device. The tasks will be sent to the task scheduler, which uses the CA to allocate them to the suitable VMs. Once the tasks are processed, the results are returned to the users. Every user has the ability to generate an unlimited number of tasks from any device. Each Task possesses distinct quantities of instructions, data size, and placement. Users submit tasks to the cloud system as input for the task scheduler. The Task Scheduler is regarded as the primary mechanism for load balancing and resource allocation. In this case, CA will manage Task Scheduling by utilizing a cost function that aims to minimize both the make span and the overall time of the tasks. The make span is determined through the following equations [24, 27]:

$$Max(FT) \quad (1)$$
$$ft_{ij} = ft_{i-1j} + PT_i$$
$$PT_i = NIT_i \times VMS_j$$

Where:

- FT is a matrix with dimensions m×n that contains the finish times of tasks.
- ft_{ij} represents the finish time of task i on VM j.
- PT_i represents the processing time of task i.
- NIT_i represents the number of instructions of task i. VMS_j represents the speed of VM j.

The total time is computed using the following formula:

$$TT_k = PT_k + ST_k \quad (2)$$

where TT_k is the processing time of task k,

PT_k is the processing time of task k and ST_k is sending time of task k to a VM.

And the cost function will be:

$$Minimize(Max(FT), TT_k) \quad (3)$$

The suggested Task scheduler will follow the following steps:

Step 1: Arrange tasks in a systematic manner within the task list.

Step 2: Utilize the CA algorithm to determine the optimal assignment of VMs based on the cost function defined in Equation (3).

Step 3: Dispatch assignments to the selected virtual machines.

Step 4: In the event that new tasks are introduced, include them in the task list and go to Step 1.

The task scheduler determines the most efficient virtual machine (VM) to execute a task based on the task's requirements. Each host has many VMs that handle task processing, with each VM having varying memory sizes and processing capacity allocated from the host.

6 Experimental environment and results

The experimental environment of cloud computing refers to the creation and organization of a cloud computing system that is especially tailored for the purpose of performing tests and experiments [23]. This environment utilized by authors, developers, and IT professionals in order to study and evaluate the cloud performance, scalability, dependability, and security, see figure 2

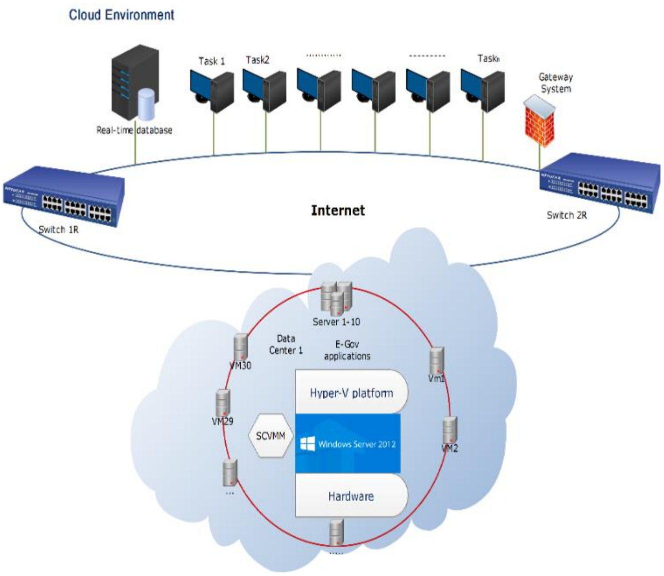

Figure 2: Experimental environment for cloud computing

As shown in Figure 2 and Table 2, the cloud experimental environment typically consists of the fallowing components: -

- **Virtual machines (VMs):** it computer programs or software that emulate a physical computer and allowing users to run several applications on a single physical machine.
- **Cloud services:** Software as a Service (SaaS) refers to the services offered by the cloud provider and it encompass storage, messaging, database, and computing services. These services are utilized for constructing and implementing cloud-based applications.
- **Networking:** The host server use has reached a bandwidth transformation of 248 Kbps. To enable communication between virtual machines (VMs) and cloud services, this includes setting up virtual networks, subnets, and security groups.
- **Monitoring and management tools:** The chosen physical server for hosting the cloud environment is equipped with an Intel(R) Xeon(R) CPU E3-1225 V5 processor running at 4.30GHz, 32GB of RAM, and a 4TB hard disk. As depicted in Fig2, the cloud infrastructure has been constructed with MSCM 2012 R2 (MSCM) and the Hyper-V hypervisor. These tools are utilized to oversee the efficiency and well-being of the cloud infrastructure, as well as to control the setup and implementation of resources.

Table 2: Cloud environment components

Name	Parameters	Value
Cloud-Datacenter	No. of Hosts	6
	No. of Data center	3
VMs	No. of VMS	30
	No.of PE per VM	1-12
	RAM	512-2200
	Bandwidth	1000-10000
Task	Total number of task	10-100
	Length of task	10-1000

Using the above real-life scenario, CA has been implemented on 64 tasks of different sizes and 30 Virtual Machines (VMs) with diverse capabilities. The cost function utilized in this situation is the duration of processing. Figure 3 presents the outcomes of the CA scheduler and it displays the cumulative processing time of the tasks in 10 distinct iterations for the CA, which is subsequently contrasted with the RR Total processing time. The results demonstrate that the proposed solution surpasses the RR time (RR_time=43088 milliseconds) in terms of both the minimum time (MIN=1980 milliseconds) and the average time (AVG=2063.6 milliseconds).

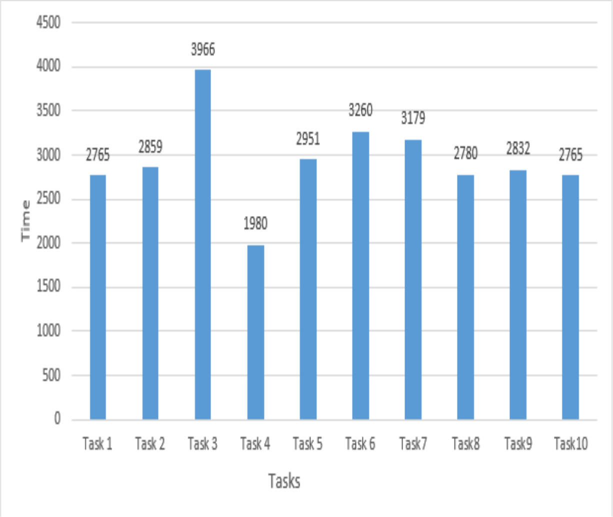

Figure 3: Total processing time In (Ms) for 10 tasks

From above results, the algorithm can be tailored to optimize specific objectives, such as minimizing task completion time, mitigating energy consumption, or balancing resource loads. Also, CA exhibits simplicity, parallelizability, and effectiveness in handling optimization difficulties with multiple objectives. It demonstrates good global search capabilities, making it suitable for complex and dynamic environments. However, fine-tuning algorithmic parameters and ensuring convergence in certain scenarios remain challenges. Additionally, the balance between exploration and exploitation needs to be carefully managed for optimal performance.

7 Discussion

In this true situation, a dataset comprising 64 positions of various sizes and 30 Virtual Machines (VMs) with various abilities is harnessed to evaluate the use of the insect settlement calculation (CA) for task planning for a distributed computing climate. Streamlining task planning for understanding with the expense capability of handling length is the primary objective. The discoveries are shown in Figure 2, which contrasts the general handling time accomplished and the Cooperative effort (RR) planning procedure to the aggregate handling season of undertakings north of ten unique stresses of the subterranean insect state technique. For this situation, the subterranean insect province strategy is utilized to progressively allocate occupations to virtual machines (VMs) as per the responsibility circulation and each machine's handling limit. This strategy is planned to increment task booking effectiveness. Insect settlement calculation: this calculation utilizes the possibility of aggregate knowledge to upgrade work assignment choices. It is enlivened by the scavenging conduct of subterranean insects. This is as opposed to regular booking strategies like Cooperative effort, which circulate occupations in a repeated way without representing the handling needs of each undertaking or the state of the virtual machines at that point. Assignments are gradually given to virtual machines (VMs) through iterative redundancies of the subterranean insect province calculation, which depends on pheromone trails that show the allure of undertaking VM pairings. Pheromone focuses are adjusted as the calculation repeats in light of the type of errand VM allotments, which ultimately brings about the progress of more compelling undertaking booking arrangements. The subterranean insect settlement calculation's versatile nature empowers it to improve task handling spans and increasingly adjust to moving responsibility conditions. A significant method for understanding the presentation acquires made by the insect province calculation is to contrast it and the Cooperative booking technique. Despite the fact that cooperative booking gives a clear and unsurprising strategy for dispensing errands, it could end in under ideal results, particularly in circumstances where assignment needs and virtual machine capacities are not uniform. Then again, the subterranean insect state strategy beats Cooperative planning as far as both least and normal handling times due to its ability to adaptively streamline task distributions in view of dynamic criticism. The subterranean insect settlement calculations noticed least handling time (MIN) of 1980 milliseconds and normal handling time (AVG) of 2063.6 milliseconds show how well it attempts to diminish task fulfillment times and lift framework productivity overall. These discoveries feature how nature-motivated calculations, like the subterranean insect state calculation, can be more successful and adaptable than ordinary booking methods in taking care of muddled advancement issues in distributed computing settings.

The flexibility and efficiency of state-line-computing can be gleaned from the fact that it could be used to execute tasks in distributed computing. The flexibility of CA for certain needs, its capacity to save energy and last but not the least, the appropriate adjustment height in accordance with the time or cycle is mainly one of its characteristics. The code can be tailored to the different requirements of companies. CA is distinguished by its simplicity, which contributes to smooth implementation and implementation of framework. Being diverse, it can facilitate solving complicated development issues. CA has potential to effectively deal with developmental issues, which have several targets. CA should look at certain scenarios which can aid in finding out a Pareto optimal plan. Applicable to instances when a lot of factors need to be kept in mind concurrently, like categorization. As a result, CA can execute global monitoring to investigate the configurations and find appropriate plans in various parts of the world. However, this solution is not perfect and often there is a lack of security and overcome of environment restrictions. Two controlling factors are the effect of pheromone diffusion and pair restriction on the outcome of CA. However, choosing the most appropriate option can be challenging and requires considerable trial and error.

Furthermore, drilling the right combination which will lead to a win is the issue as well. There is also the issue of finding this balance between inquiry and discussion. The purpose of the second meeting is to have the current regulations revised and improved which were however discovered that new regulations were introduced, probably even better ones. We have to identify and address these two divergence points to get the best results and to reduce the risks of poor planning and conflicts. Controlling activities in individual areas of the state that use pesticides is a good way to achieve many goals, including limiting the use of the property and when it can be done. It is a tool used in solving development problems due to its simplicity, ease and ability to solve many purposes. However, to achieve maximum potential when using similar content, issues such as linking and transfer need to be addressed appropriately. Overall, CA is important for many development strategies with good results and has the potential to improve the feasibility and success of the classification process.

In a comparative path, for scheduling 64 tasks on 30 virtual machines (VMs), the CA scheduler was able to achieve a minimum processing time of 1980 milliseconds and an average processing time of 2063.6 milliseconds across 10 iterations. This shows a notable improvement over the 43088-millisecond overall processing time of the round-robin (RR) scheduler. These outcomes are on par with or even superior to previous research on task scheduling that included bio-inspired optimization methods, such as ant colony optimization (ACO) and particle swarm optimization (PSO). In contrast to a simple scheduler, Tawfeek et al.'s [10] ACO-based cloud workflow scheduler produced makes pan savings of up to 29%. A discrete PSO technique for cloud workflow scheduling was proposed by Wu et al. [28]; on average, this algorithm reduced makes pan by over 15%. Goals like cutting down on completion times and efficiently distributing loads among virtual machines are optimized by the CA scheduler. This is consistent with the findings, who found that, in comparison to other heuristic algorithms, genetic algorithm-based scheduling lowered task processing times by as much as 47%. The CA scheduler's performance is contingent upon parameter adjustment and meeting convergence conditions. In order to avoid early convergence, more research could concentrate on self-adaptive parameter control and diversity maintenance. All things considered; the CA scheduler exhibits potential as a productive metaheuristic method for scheduling tasks in cloud data centers. Table 3 summarizes the performance metrics task processing time, resource utilization, scalability, and load balancing average for 64 tasks comparing with studies mentioned in Section III. The comparison utilizes the same cloud configuration settings have been based in this study.

Table 3: An analysis of performance metrics across 8 distinct experiments

Authors /Ref	Task processing time (ms)	Resource utilization (%)	Scalability (10000 workload)	Load balancing average (%)
Our Solution	2063.6	33	0.5265	95.13
Moradbeiky and Bardsiri [12]	2313	27	0.3899	67.2
Abadi, Z. J. K., and Mansouri, N. [14]	2500.6	25.4	0.3232	72.1
Ghafari, R., et al. [15]	2667.11	23.7	0.4539	53.8
Murad, S. A. et al. [16]	2089.2	32	0.1931	84.4
Walia and Kaur's study [17]	2263	29.7	0.4899	73.7
Ajmal, M. S. et al. [18]	2543.7	26.6	0.4421	71.2
Kashikolaei et al.[19]	2403.6	28.5	0.3899	56.3

Table 3 demonstrates that the discussion of CA scalability becomes more significant as task workloads above 10000. The authors chose this particular value as a sample in order to assess the scalability of CA. The first execution time is dependent on the setting of the element and its size in the cloud virtual machine. CA facilitates the functioning of the cloud datacenter by allocating the job workload to virtual machines and adjusting it dynamically when the size of the element increases. Table 3 demonstrates that the CA exhibits superior scalability performance by yielding a satisfactory value of 0.5265 in comparison to alternative scheduling algorithms. All the results were examined in an actual cloud computing environment with identical configurations, settings, and task contents.

8 Research gap and limitations

Certainly, even while task planning expansion for cloud datacenters has progressed fundamentally, there is as yet a sizable exploration hole in the creation and use of shrewd booking calculations that are mainly fit to manage the particular obstacles introduced by distributed computing conditions. Current booking calculations could not enough exploit the difficulties and dynamism of cloud datacenters since they habitually depend on heuristic or metaheuristic techniques. In addition, a deficiency of careful examinations exists that deliberately survey how well clever planning calculations capability in genuine cloud frameworks while considering factors like asset use proficiency, versatility, and adaptability to evolving jobs. To close this analysis hole, imaginative wise booking calculations that consolidate computerized reasoning, AI, and high level improvement approaches should be planned and assessed to create errands that are both adaptable and proficient preparation in distributed computing offices. Observational examination is likewise expected to assess the reasonability and viability of these calculations in upgrading framework execution and asset use across a scope of distributed computing situations. Consequently, regarding to limitations associated with our prosed solution, below the main points of CA implementation in cloud computing:

- Cuckoo search calculations (CA) can be rigid to perform for work planning for cloud datacenters; it needs an intensive understanding of the boundaries and distinctions of the calculation.
- While working with broad scattered computing frameworks that have innumerable exercises and virtual machines, CA's versatility might be restricted.
- With restricted memory and controlling power, cloud datacenters might find it difficult for CA to streamline work booking.
- It tends to be tough to ensure combination to an ideal organization, particularly in responsibility conditions that are vibrant and surprising.
- It can need a great deal of investment and trial and error to calibrate algorithmic variables like the step size and examination pace.
- On the off chance that CA isn't enough instated, its exhibition could be helpless to the underlying circumstances and arrangements, coming about in under ideal arrangements.
- Over fitting the model to a definite dataset or situation conveys a gamble of delivering arrangements that are not adaptable to other distributed computing conditions.
- The compromise among investigation and double-dealing in CA may not generally be adjusted, which could bring about either excessively exploratory assembly.
- The worldwide ideal arrangement may not generally be found by CA, and it sometimes gets caught in nearby optima.
- While involving CA for task arrangement, there might be a huge computational expense included, principally in high-through put or constant settings.

9 Conclusion and future works

Authors regularly focus on improving the pertinence of the Cuckoo Search Calculation in task booking, joining it with different calculations, and thinking about its exhibition in contrast to standard difficulties. The calculation's adaptability and bio-propelled approach add to constant endeavors to work on the proficiency of cloud datacenter activities. The CA makes a huge commitment to the field of clever calculations by harnessing an original avian-enlivened procedure to streamline work planning for cloud datacenters. Further analysis and advancement of CSA's latent upgrades are supposed to add to extra advance in the field of improvement calculations. Also, this study leads an essential calculation of the heap adjusting issue in distributed computing, zeroing in on wise streamlining systems. It gives a brief explanation of generally utilized methods, like hereditary calculation (GA), subterranean insect state streamlining (ACO), counterfeit honey bee settlement (ABC), and molecule swarm improvement (PSO). These calculations display better execution analyzed than customary ones as far as administration quality by bringing down response time and tending to makes skillet concerns. Nonetheless, they are tormented by network above and nearby enhancement issues, bringing about defers in both the undertaking task cycle and errand handling time, as well as inefficient asset usage. Afterward, the exploration recommends a clever insightful methodology for work booking that utilizes the CA calculation. The proposed approach shows critical likely as far as execution during the concise reenactment adjusts contrasted with the Cooperative effort (RR) calculation. Moreover, when contrasted with other insightful calculations intended for improvement, it is expected to display a critical level of investigation inside a hunt space. Moreover, it is projected to really oversee broad and differed search spaces, especially in a dynamic and versatile climate, for example, distributed computing.

At the future, a lot of optimizations should be achieve and implement on the proposed scheme through implementing it in the public cloud server and develop a new layer in cloud structure naming as "Smart Scheduling as a service".

References

[1] Farrag, A. a. S., Mahmoud, S. A., & El-Horbaty, E. S. M, . *Intelligent cloud algorithms for load balancing problems: A survey*. 2015, 10.1109/IntelCIS.2015.7397223. https://doi.org/10.1109/intelcis.2015.7397223

[2] Mohammad, O. K. J. , *GALO:A new intelligent task scheduling algorithm in cloud computing environment*. International Journal of Engineering & Technology, 7(4), 2088. 2018, https://doi.org/10.14419/ijet.v7i4.16486

[3] Priya, & Babu, C. N. K., *Moving average fuzzy resource scheduling for virtualized cloud data services*. Computer Standards & Interfaces, 50, 251–257. 2018, https://doi.org/10.1016/j.csi.2016.10.011

[4] R. Govindarajan, S. Meikandasivam, and D. Vijayakumar, "Performance Analysis of Smart Energy Monitoring Systems in Real-time", Eng. Technol. Appl. Sci. Res., vol. 10, no. 3, pp. 5808–5813, Jun. 2020. https://doi.org/10.48084/etasr.3566

[5] Zeng, L., Veeravalli, B., & Zomaya, A. Y., *An integrated task computation and data management scheduling strategy for workflow applications in cloud environments.* Journal of Network and Computer Applications, 50, 39–48. 2015, https://doi.org/10.1016/j.jnca.2015.01.001

[6] Keshk, A., El-Sisi, A. B., & Tawfeek, M. A., *Cloud Task Scheduling for Load Balancing based on Intelligent Strategy.* International Journal of Intelligent Systems and Applications, 6(5), 25–36, 2015, https://doi.org/10.5815/ijisa.2014.05.02

[7] Pradhan, R., & Satapathy, S. C., *Particle Swarm Optimization-Based Energy-Aware task scheduling algorithm in heterogeneous cloud.* In Lecture notes in networks and systems (pp. 439–450), 2022, https://doi.org/10.1007/978-981-19-4990-6_40

[8] Varghese, B., & Buyya, R., *Next generation cloud computing: New trends and research directions*. Future Generation Computer Systems, 79, 849–861, 2018, https://doi.org/10.1016/j.future.2017.09.020

[9] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. a. F., & Buyya, R., *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms*. Software: Practice and Experience, 41(1), 23–50, 2010, https://doi.org/10.1002/spe.995

[10] Tawfeek, M. A., El-Sisi, A. B., Keshk, A., & Torkey, F. A., *Cloud task scheduling based on Ant Colony optimization.* The International Arab Journal of Information Technology, Vol. 12, No. 2, March 2015, https://doi.org/10.1109/icces.2013.6707172

[11] Abrishami, S., Naghibzadeh, M., & Epema, D., *Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds*. Future Generation Computer Systems, 29(1), 158–169. 2018, https://doi.org/10.1016/j.future.2012.05.004

[12] Moradbeiky, A., & Bardsiri, V., *A Novel Task Scheduling Method in Cloud Environment using Cuckoo Optimization Algorithm*. IJCS, 2(2), 7–20, 2014, https://doi.org/10.21742/ijcs.2015.2.2.02

[13] Vijindra, & Shenai, S., Survey on scheduling issues in cloud Computing. Procedia Engineering, 38, 2881–2888, 2012, https://doi.org/10.1016/j.proeng.2012.06.337

[14] Abadi, Z. J. K., & Mansouri, N., *A comprehensive survey on scheduling algorithms using fuzzy systems in distributed environments.* Artificial Intelligence Review, 57(1), 2024, https://doi.org/10.1007/s10462-023-10632-y

[15] Ghafari, R., Kabutarkhani, F. H., & Mansouri, N., *Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review.*

Cluster Computing, 25(2), 1035–1093, 2022, https://doi.org/10.1007/s10586-021-03512-z

[16] Murad, S. A., Muzahid, A. J. M., Azmi, Z. R. M., Hoque, M. I., & Kowsher, M., *A review on job scheduling technique in cloud computing and priority rule based intelligent framework.* Journal of King Saud University - Computer and Information Sciences, 34(6), 2309–2331, 2022, https://doi.org/10.1016/j.jksuci.2022.03.027

[17] Walia, N. K., & Kaur, N., *Performance analysis of the task scheduling algorithms in the cloud computing environments*. 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 2021, https://doi.org/10.1109/iciem51511.2021.9445320

[18] Ajmal, M. S., Iqbal, Z., Khan, F. Z., Ahmad, M., Ahmad, I., & Gupta, B. B., *Hybrid ant genetic algorithm for efficient task scheduling in cloud data centers*. Computers & Electrical Engineering, 95, 107419, 2021, https://doi.org/10.1016/j.compeleceng.2021.107419

[19] Kashikolaei, S. M. G., Hosseinabadi, A. a. R., Saemi, B., Shareh, M. B., Sangaiah, A. K., & Bian, G., *An enhancement of task scheduling in cloud computing based on imperialist competitive algorithm and firefly algorithm*. The Journal of Supercomputing, 76(8), 6302–6329, 2019, https://doi.org/10.1007/s11227-019-02816-7

[20] Tang, C., Song, S., Ji, J., Tang, Y., Tang, Z., & Todo, Y., *A cuckoo search algorithm with scale-free population topology*, Expert Systems With Applications, 188, 116049, 2021, https://doi.org/10.1016/j.eswa.2021.116049

[21] Rajabioun, R., *Cuckoo Optimization Algorithm*, Applied Soft Computing, 11(8), 5508–5518, 2012, https://doi.org/10.1016/j.asoc.2011.05.008

[22] Sen, S., Li, J., Huang, Q., Xh, H., Shuang, K., & Jie, W., *Cost-efficient task scheduling for executing large programs in the cloud*. Parallel Computing, 39(4–5), 177–188, 2013, https://doi.org/10.1016/j.parco.2013.03.002

[23] Swathi Velugoti, M. P. Vani, AN APPROACH FOR PRIVACY PRESERVATION ASSISTED SECURE CLOUD COMPUTATION, Informatica, 47, 41–52, 2023, https://doi.org/10.31449/inf.v47i10.4586

[24] M. Ali, N. Q. Soomro, H. Ali, A. Awan, and M. Kirmani, “Distributed File Sharing and Retrieval Model for Cloud Virtual Environment”, *Eng. Technol. Appl. Sci. Res.*, vol. 9, no. 2, pp. 4062–4065, Apr. 2019. https://doi.org/10.48084/etasr.2662

[25] Wu, Z., Ni, Z., Gu, L., & Liu, X., *A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling,* 2010 International Conference on Computational Intelligence and Security, CIS 2010, 2010, https://doi.org/10.1109/cis.2010.46.

[26] H. Reffad, A. Alti, and A. Almuhirat, “A Dynamic Adaptive Bio-Inspired Multi-Agent System for Healthcare Task Deployment”, Eng. Technol. Appl. Sci. Res., vol. 13, no. 1, pp. 10192–10198, Feb. 2023. https://doi.org/10.48084/etasr.5570

[27] Fouad H. Awad, Murtadha M. Hamad, Big Data Clustering Techniques Challenges and Perspectives: Review, Informatica, 47, 203–218, 2023, https://doi.org/10.31449/inf.v47i6.4445