International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/221890 Volume 10, Issue 3, May – June (2023)

RESEARCH ARTICLE

A Hybrid Secure and Optimized Execution Pattern Analysis Based O-HMACSHA 3 Resource Allocation in Cloud Environment

Himanshu
Department of Computer Science and Engineering, MMEC, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala, Haryana, India.
gupta91.himani@gmail.com

Neeraj Mangla
Department of Computer Science and Engineering, MMEC, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala, Haryana, India.
erneerajynr@gmail.com

Received: 13 April 2023 / Revised: 29 May 2023 / Accepted: 07 June 2023 / Published: 30 June 2023

Abstract – According to the analysis, several task scheduling methods have been implemented, such as the Particle Swarm Optimization (PSO) method, which has enhanced the performance of cloud data centers (DCs) in terms of various scheduling metrics. The scheduling issue in cloud computing (CC) is well-known to be NP-hard, with the main challenge arising from the exponential increase in the no. of possible outcomes or groupings as the problem size grows. Therefore, the key aim is to determine secure and optimal solutions for scheduling consumer tasks. In this study, a proposed method called Optimized-Hybrid Medium Access Control Secure Hash Algorithm 3 (O-HMACSHA3) is introduced for CC. The investigation aims to address the issue of reliable resource scheduling access for different tasks in the cloud environment, with a focus on reducing turnaround time (TAT) and energy consumption (EC). The proposed method utilizes optimization with PSO to achieve soft security in resource scheduling. To evaluate its effectiveness, the research method is compared with other task scheduling methods, including PSO and Fruit Fly-Based Simulated Annealing Optimization (FSAO) method, in terms of EC and time. The findings indicate significant improvements in performance metrics, with energy consumption reduced to 47.7 joules and TAT decreased to 316 milliseconds compared to existing methods. This is in contrast to the proposed method, which resulted in 57.3 joules and 479 milliseconds, respectively, for 20 tasks.

Index Terms – Task Scheduling (TS), O-HMACSHA3 (Optimized-Hybrid Medium Access Control Secure hash Algorithm), PSO (Particle Swarm Optimization), EC (Energy Consumption), TAT (Turnaround Time).

1. INTRODUCTION

Currently, technological advancements have given rise to a new concept called cloud computing. It enables the provision and hosting of facilities and resources to numerous users via the Internet on a pay-as-you-go basis [1]. Through various cloud computing (CC) vendors, consumers can access networking, computer infrastructure, data storage, and other applications as needed to fulfill their requirements. It encompasses dissimilar models, such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), to offer these cloud-based services. In this context, the classical service provider role can be classified into two leading points.

- Organization companies handle cloud stages and offer resources using a procedure-based assessing scheme.
- Cloud service providers rent resources from any organization provider to meet customer requests.

Managing cloud services is challenging due to their dynamic, diverse, and self-configuring nature. The key concerns of task scheduling (TS) [2] in CC are load balancing (LB) and resource management. On a multiprocessor system, TS becomes an NP-complete issue. For this reason, several investigators have implemented various heuristics and meta-heuristic methods to address this problem. Different categories of these heuristic methods include scheduling and clustering methods. They analyze and select optimization methods for TS in the cloud from a large pool of heuristic and meta-heuristic methods [3].

It is an on-demand platform that provides clients with access to collective resources, software, and other equipment at pre-determined times. This term is frequently used in the context of the Internet, where the entire Internet can be seen as a cloud. It can help lower capital and effective costs.

RESEARCH ARTICLE

Automatic LB services, which are the foundation of cloud vendors, allow organizations to scale up the number of CPUs or memory in their systems to meet growing demands. Depending on the business needs of the entity, this service is optional. Load balancing fulfills two crucial functions: promoting the availability of cloud services and enhancing performance. Figure 1, defined in network diagrams, illustrates a cloud used to represent the Internet [4].

This rapid development places a focus on Cloud Service Providers (CSPs) to evaluate effective resource management. In addition, it is necessary to distribute client demands among Virtual Machines (VMs) to handle more requests without expanding the physical infrastructure [5]. As a result, LB and TS are central themes in CC and have recently increased important care. For instance, general IaaS cloud provider AMAZON EC2 (elastic compute cloud) utilizes elastic LB to allocate client demands [6]. It is important to note that load balancing (LB) and TS for cloud computing are NP-hard issues.

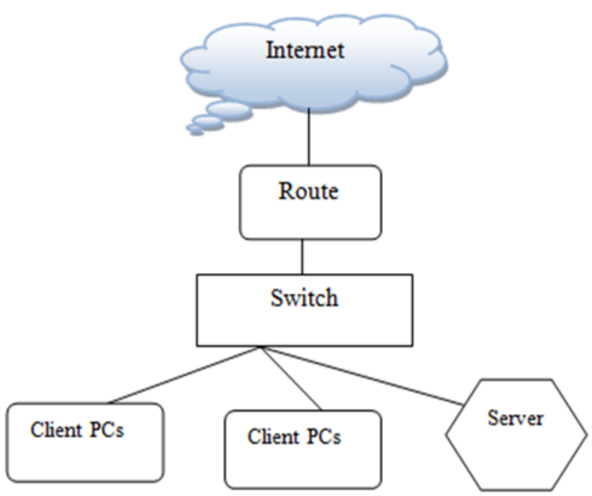

Figure 1 General Representation of Cloud Computing [4]

Additionally, in addition to load balancing, Cloud Service Providers (CSPs) must meet Quality of Service (QoS) necessities such as flexibility, availability, and throughput to attract consumers [7][8]. Due to the diverse environments in CC, numerous user requests are available for execution. Through an IaaS broker, users input their resource requests. In certain cases, users may have unique needs that require an agreement, called as a Service Level Agreement (SLA), where CSPs must provide guaranteed services. In the context of CC, user requests are often referred to as tasks. Various task scheduling methods are employed to execute these tasks on the obtainable cloud data center machines [9]. The main goal of TS methods is to focus on different QoS parameters, such as system throughput and response time. The cloud environment allows for the allocation of appropriate resources to meet the task requirements, including those for the Operating System (OS), memory, and time. Task scheduling methods may increase the time essential to whole a task and reduce the system throughput of the cloud. Therefore, the aim is to enhance the overall performance and utilization of these services in different environments [10]. Various TS methods are used in cloud computing environments, such as Ant Colony Optimization (ACO) [11], Bee Colony Optimization (BCO), and Genetic Algorithm (GA) [12].

The chief involvement of the propose work is to develop secure and optimized resource scheduling access for different jobs in the cloud environment. The PSO method has been employed to ensure secure and improved resource scheduling access for multiple tasks. This method has addressed existing issues and improved performance metrics related to energy consumption and turnaround time.

1.1. Objectives

To design soft security for reliable resource scheduling access for different tasks in the CC environment.

The research paper is arranged as trails: Section 2 defines various TS methods for CC and discusses optimization and encryption techniques. The problem definition is offered in Section 3. Section 4 defines the various classifications of task scheduling techniques in cloud computing. Section 5 explores meta-heuristic and heuristic scheduling techniques. The research proposal is elaborated in Sections 6 and 7, which cover the experimental outcomes, conclusion, and upcoming improvements.

2. RELATED WORK

Numerous recent analysis have been showed in the field of CC, focusing primarily on areas such as task development, virtual machines (VMs), evaluation, load balancing (LB), and energy management. LB, in particular, has been a focal point for analysts due to its significance in CC for stakeholders including cloud service sources and users. Based on the analysis of previous literature surveys, one of the details for this attention is the lack of comprehensive classification between numerous LB methods. In this division, a systematic examination of earlier research works is presented. According to the author [13], in their proposed work, various task scheduling methods such as GA, ACO, and others were implemented, leading to improved performance of cloud data centers (DCs) in terms of several task scheduling metrics. These scheduling issues are known to be NP-hard, as the no. of possible answers increases exponentially with the problem size. Therefore, achieving optimized scheduling of consumer jobs is a challenging task. This research implemented an adaptive LB task scheduling method for CC. This method maps external tasks to obtainable VMs in a load-balanced manner to optimize the make span, increase resource utilization (RU), and adaptively reduce Service Level

RESEARCH ARTICLE

Agreement (SLA) violations. The proposed TS approach's performance was measured and compared with other optimization approaches using metrics such as average RU; make span, and SLA violation. The author [14] described research on optimization algorithms such as resource allocation, load balancing (LB), non-linear objective function (OF) optimization, queuing cost, cloud cost estimation, and task particle optimization methods for scheduling in a cloud environment. The key aims of the research are as follows:

- To implement an effective task-based dynamic load-balanced distributed queue that maps tasks to resources, optimizing cost and execution time while improving fault tolerance and reliability.
- To provide a multi-objective optimization (MOO) based strategy for VM consolidation that considers fault tolerance (FT), load balancing (LB), job priority, and efficient resource allocation.
- To achieve an improved relocation evaluation structure and efficient networking, memory, and TS requirements modeling.

Harvinder Singh et al. (2021) [15] proposed addressing the workflow scheduling issue with the goal of maximizing resource utilization. They implemented QoS-based resource allocation and TS using swarm-based ACO method to achieve more expected outcomes and reduce scheduling optimization issues. The implemented methods were simulated in a simulated cloud environment. In future work, the effects of the implemented method will be compared with other policies and evaluated using QoS metrics. Muhammad et al. (2021) [16] discussed an effective adaptive migration method developed for operational migration and placement of VMs on physical machines (PMs). The research method consists of two components: (i) selecting PM positions with the best access latency for reducing VMs and (ii) optimizing the no. of VM migrations. The research was extensively simulated using the Cloudsim toolkit. The outcomes of the research method were compared with proactive simulation-based scheduling, load balancing (LB), and task-aware scheduling heuristic techniques in terms of SLA violation, RU, no. of hosts shut down, and EC. The results show that the effective adaptive migration method expressively decreases the no. of migrations by 16% and 24%, SLA violation by 20% and 34%, and maximizes RU by 8% to 17%. Additionally, it reduces the maximum no. of hosts shut down from 10% to 30% compared to the existing methods. Shabaz et al. (2019) [17] discussed the problem of load imbalance as a multi-variant and constraint issue that negatively affects the efficiency and performance of computational systems. Load balancing (LB) methods offer solutions for addressing load imbalance conditions, focusing on both under loading and overloading scenarios. The article presents a research analysis of LB techniques, highlighting the benefits and challenges of existing methods and identifying critical limitations that need to be addressed to further develop effective load-balancing methods. Additionally, the article proposes novel insights into LB in cloud computing. Mohit et al. (2018) [18] introduced a reliable cloud framework for efficiently managing high consumer demands while meeting deadlines and ensuring resistance to failures using a threshold-based trigger plan. The evaluation outcomes demonstrate that the implemented method significantly decreases the make span time and increases the task acceptance ratio by over 10% compared to the min-min approach and by 20% compared to FCFS (first-come, first-served) and SJF (shortest job first) algorithms in all scenarios. Danlami et al. (2022) [19] discussed an implemented method where virtual annealing was incorporated to balance local and global search and reduce premature convergence. They introduced a trade-off parameter that allows users to choose a reliable Quality of Service (QoS) level that minimizes the cost of execution. The FSAO method was designed and executed using the EDGECLOUDSIM simulation tool. The experimental results demonstrate that the FSAOS efficiently schedules resources based on task requirements, reducing the make span and cost of execution. It achieves better RU compared to the conventional FFO (fruit-fly optimization) and PSO methods. Table 1 provides a description of existing research methods, problems, tools, and metrics related to LB and TS in cloud computing.

Table 1 Existing Analysis of the Different Methods for Cloud Computing

Year	Methods	Problems / Gaps	Tool / Server	Parameters
2021 [13]	ALTS GA ACO	Imbalance mapping of tasks	CloudSim	Makespan ARUR SLA violation
2020 [14]	max-min min-min	Minimum makespan scheduling issues, memory storage	AWS	Start time Finish time Time

RESEARCH ARTICLE

2021 [15]	ACO	NP-hard issues Traveling salesman issues	Cloud tool	R^2 values Execution cost RU
2021 [16]	EAMA	Due to consuming high energy consumption	CloudSim	RU EC SLA violation No. of migrations Number of host shutdown
2019 [17]	Hierarchical taxonomical classification	Load unbalancing NP-Hard problem Overloading problem	C++ and C CloudSim MATLAB Cloud analyst	RT ET RU Makespan Scalability Execution cost
2018 [18]	FCFS SJF	Deadline Less priority	Cloud analyst Eucalyptus	Makespan TAR
2022 [19]	FSAOA	Optimization issues Resource allocation issue	Edge CloudSim	Makespan time Execution cost

The classification in Table 2 is based on TS and various performance metrics derived from current research articles. In the table, the presence or absence of parameters is indicated by 'Yes' or 'No', respectively.

Table 2 Comparison Analysis of Task Scheduling Methods with Parameters

Methods used in task scheduling	ARUR	Cost	EC (Execution Cost)	Makespan
FASOS [19]	No	No	Yes	Yes
I-PSO [20]	Yes	Yes	No	Yes
I-CGA[21]	Yes	No	Yes	No
PSO[22]	Yes	No	No	Yes
L-BACO[23]	Yes	No	Yes	Yes
I-CDFS[24]	Yes	No	Yes	Yes
ACO[25]	Yes	No	Yes	Yes
GA-ACO[26]	Yes	Yes	Yes	Yes

3. PROBLEM DEFINATION

Cloud computing revolutionizes a significant portion of the IT (Information Technology) sector, transforming the concept of data processing into a valuable asset. It impacts the development and consumption of equipment, increasing the popularity of SaaS and reshaping the industry. The appeal of CC lies in its ability to lease computational resources on-demand, making it an increasingly attractive option for users. Cloud hosting operates on an on-demand or pay-as-you-go model, granting customers the perception of infinite computational resources in a shared environment. These resources can be scaled up or down to meet the necessities of cloud users, necessitating precise task scheduling for optimal utilization.

Table 3 Current Methods with Research Gaps

Year	Proposed techniques	Research gaps
2021[13]	Adaptive load balancing based task management	High utilization of energy
2021 [27]	Improved Multi-Verse optimization technique	Issues in resource management
2021[28]	Genetic algorithm-based task handling	Re-transmission data issues
2021[29]	Load balancing-based technique for cloud computing	Complexities occur whenever transmitting a large volume of data
2020[30]	Hybrid technique for task scheduling	Tools for resource management are limited

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/221890 Volume 10, Issue 3, May – June (2023)

Despite the existence of several proposed techniques, cloud computing still faces challenges such as high energy utilization, resource management issues, and limited tools for resource management. In a CC system, two distinct players are involved:

- The cloud user and the cloud delivery network. Each player has different objectives, with cloud service providers striving to maximize resource usage while cloud users aim to reduce costs while fulfilling their requirements.
- However, the lack of information exchange between cloud providers and data consumers hinders resource allocation.

Table 3 presents the previous methods of TS in cloud computing, along with the proposed methods and identified gaps.

4. TAXONOMY OF TASK SCHEDULING METHODS IN CLOUD COMPUTING

In this section, the three primary categories into which they are divided are explained: heuristic, meta-heuristic, and hybrid.

4.1. Heuristic Scheduling (HS) Technique

These methods are issue-based and provide a better presentation for a particular field of issues but minimum performance for others. Generally, it offers a specific solution for a particular area of an issue in a finite quantity of time but cannot resolve complex optimization issues. Several heuristic methods have been developed in a cloud environment that determines the workflow scheduling issue and independent tasks. Now, we have deliberated several heuristic methods by dividing them into dissimilar modules based on the important keyword in the research editorial, like; as HEFT (heterogeneous earliest finish time)[31], max-min[32], FCFS [33], SJF [34], RR [35], dynamic resource scheduling and allocation methods depend on the QoS metrics[36], etc. These methods deployed the tasks at the VM utilizing various scheduling methods and tried to reduce several rates of service metrics. Analysis of several heuristic methods along with their benefits and challenges is defined in table 4.

4.2. Meta-Heuristic Scheduling (MHS) Technique

In the past 20 years, these methods have gained significant popularity for their effectiveness in solving complex computational problems. These methods possess several beneficial characteristics, including;

- They are not specific to a particular problem.
- They are approximate and often non-deterministic.
- They proficiently discover the search space to find optimal solutions (OSs) for NP-complete issues.

Examples of such methods include PSO [37], ACO [38], GA [39], HBB-LB (honey bee behavior Load balancing) [40], BAT [41], CSO (Cuckoo Search Optimization) [42], FFA (Firefly Algorithm) [43], LO (Lion Optimization) [44], ABC (Artificial Bee Colony) [40,52], and others, as shown in Figure 2. An analysis of the PSO method, its variants, techniques, quality of service metrics, and challenges can be found in Table 7.

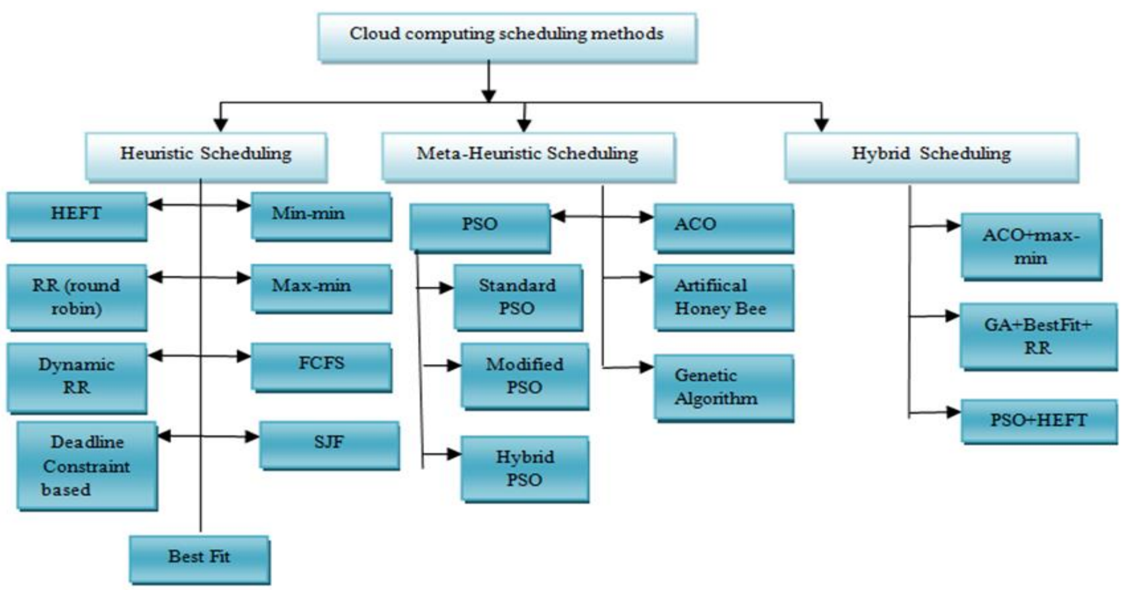

Figure 2 Taxonomy of Scheduling Methods in Cloud Computing [45]

RESEARCH ARTICLE

Table 4 Instantaneous Heuristic Methods with their Challenges and Benefits

Year	Method	Approaches	Benefits	Research Gap
2018 [46]	M-HEFT	Evaluate the rank for all tasks in DAG	Reduce the loaded issue of HEFT and the make-span time	It enables the division of the load uniformly.
2013 [47]	enhanced min-min	Task relocation method	Optimize the makespan time and rise the resource utilization	Reschedule the tasks
2014 [48]	Max-min	Assign the maximum task to best reserve initially	It optimizes the time and resource utilization	High possibility of overloaded.
2015 [49]	SJF	Choose minimum task first for implementation	It optimizes turnaround time and ET in evaluation with FCFS and RR	Load imbalanced in this method.

Table 5 QoS-Based HS Technique

Year	ET	MT (makespan time)	Execution time	RT	RU	Throughput
2018 [46]	Yes	Yes	No	No	No	No
2013[47]	Yes	Yes	No	No	Yes	Yes
2014 [48]	Yes	Yes	No	Yes	Yes	Yes
2015 [49]	Yes	No	No	Yes	No	No

Table 6 HS Technique-Based on Several Metrics along with Constraint

Year	Static	Dynamic	Resource analysis	Deadline	Priority
2018 [46]	Yes	No	Yes	No	No
2013[47]	No	Yes	Yes	No	Yes
2014 [48]	Yes	No	No	No	No
2015 [49]	Yes	No	No	No	No

Table 7 Analysis of Meta-Heuristic Methods with their Benefits and Challenges

Year	Method	Approaches	Benefits	Research Gap
2014 [37][50]	PSO	Task migration approach	Enhance the cost, time and TRR	The local minima hits Gbest, and no measurements of energy use are made.
2013 [38][51]	ACO	2-level cloud scheduler method using ACO	Reduce makespan, response time, and throughput	A secondary communication instrument is utilized to interchange the data among entities.
2013 [39]	GA	LB (load balancing) method using GA	Enhance Makespan and RT (response time)	The complexity of the technique is high.

RESEARCH ARTICLE

2018 [41]	BAT	LB-RC method using BAT	Enhance cost, RU (resource utilization), reliability, etc.	It does not describe trade-off solutions among conflict QoS metrics such as; cost and time.
2017[44]	Lion method	TS by lion	Enhance makespan time, cost, RU, and degree of imbalance	The method does not measure any constraint.
2016 [52]	ABC	Agile task managing approach using ABC	Enhance Makespan time and degree of imbalance	Critical parameters such as; energy and cost are not measured, where the deadline is a constraint.

Table 8 Quality of Service Metrics-Based MHS Method

Year	Execution time	Makespan time	Execution cost	Response time	RU	Throughput
2013 [39]	Yes	Yes	No	Yes	No	No
2018[41]	Yes	Yes	Yes	No	Yes	No
2017 [44]	Yes	Yes	Yes	No	Yes	No
2013[51]	Yes	Yes	Yes	No	Yes	Yes
2016 [52]	Yes	Yes	No	No	No	Yes

Table 9 MHS Method Based on Several Metrics Along with Constraint

Year	Static	Dynamic	Resource analysis	Deadline	Priority	FT (fault-tolerance)
2013 [39]	No	Yes	Yes	No	No	No
2018[41]	No	Yes	Yes	No	No	No
2017 [44]	No	Yes	Yes	No	No	No
2013[50]	No	Yes	Yes	Yes	Yes	No
2013 [51]	No	Yes	Yes	No	No	No
2016 [52]	No	Yes	Yes	No	No	No

An analysis of previous scheduling methods, QoS metrics, constraints, and limitations is presented in Table 5. In this table, "Yes" indicates the presence of a parameter, while "No" indicates its absence. Table 6 provides an analysis of several scheduling methods, including metrics, methods, constraints, and challenges. Several task scheduling (TS) methods for cloud environments using the meta-heuristic approach are described, along with their benefits, challenges, and algorithm performance. The details can be found in Tables 8 and 9. In these tables, the following abbreviations are used: T = True, F = False, Y = Yes, and N = No.

5. RESEARCH METHODOLOGY

Figure 3 illustrates the proposed model flowchart, which outlines the various steps to obtain the desired outcome. The process begins with the use of multiple listeners to capture user requests. These requests may contain task information of similar or different types. The listeners gather data from the Internet and organize task queues based on different priority levels. These priorities enable cloud networks to understand execution patterns and deliver high-quality services to their users. The proposed flow includes an encryption method that ensures the security of all requests and data elements. It is an enhanced version of the HMAC algorithm, incorporating the secure hash function of the SHA3-512 algorithm. This method helps generate a secure encryption key for the HMAC algorithm, resulting in superior-quality encrypted data elements.

RESEARCH ARTICLE

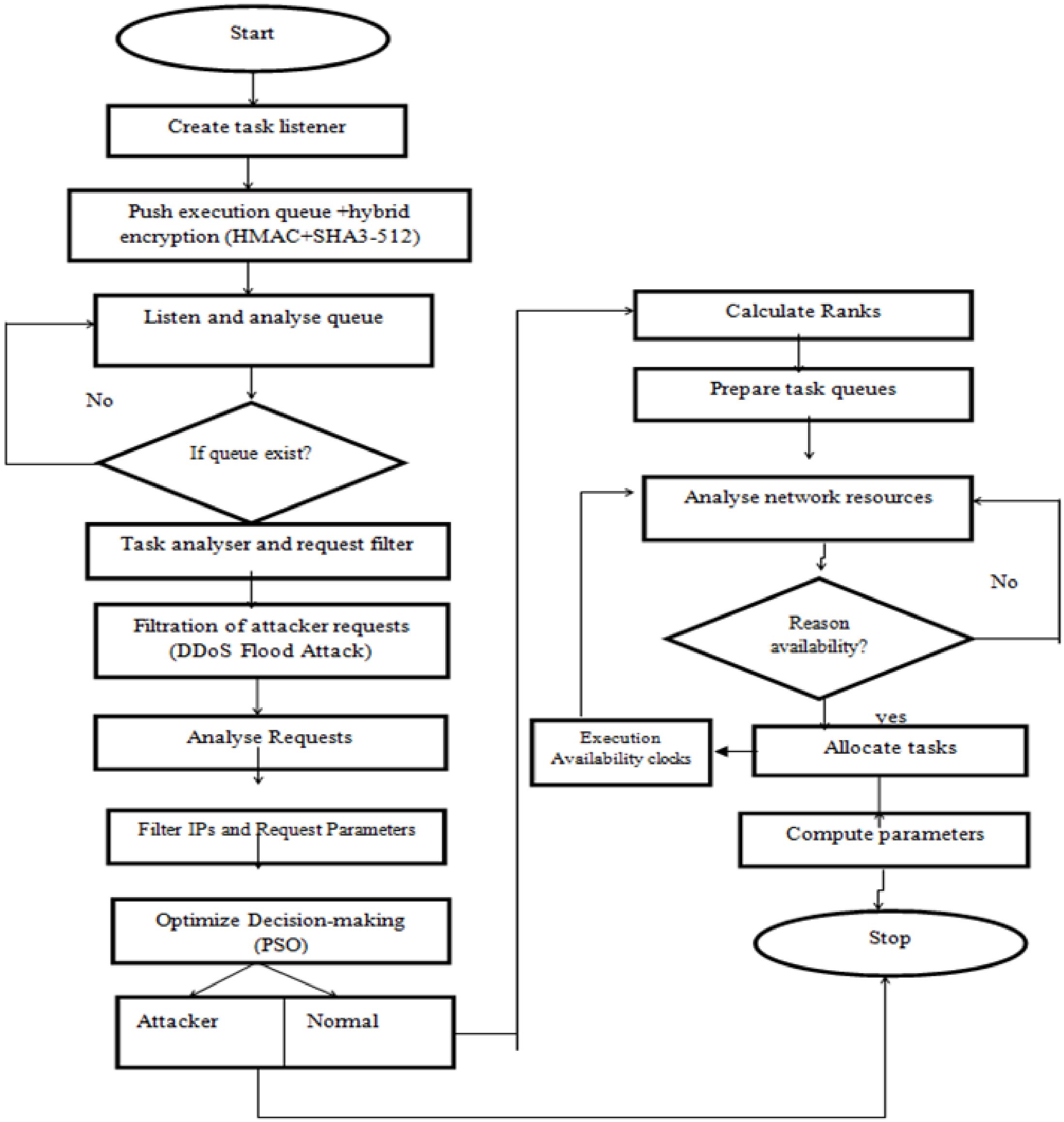

Figure 3 Flowchart of the Secure and Optimized Resource Allocation Process

The queue analyzer is a crucial component of the proposed architecture as it analyzes tasks and their header components. The analyzer verifies function properties and estimates the effort required for execution. If there is a queue that requires resources for execution, the analyzer performs request filters to identify attackers and ensure regular performance [53]. Attackers in this environment employ distributed denial of service (DDOS) attacks to bring down or crash the cloud network. Each request has different properties and a specific timespan to detect potential attackers. The proposed model includes an optimization process for analyzing request parameters and filtering out attacker requests in a cloud environment. The decision-making module of the optimization process eliminates attacker requests and blocks user IDs, while managing standard requests in execution queues processed by the cloud network. This iterative process handles resource allocations, tracks occupied resource status, and ensures resource availability based on execution clocks.

Once all processes are completed following the given flow, the performance analyzer calculates the efficiency of the developed model. Various parameters are used to obtain readings and make comparisons to validate differences. Table 10 defines the resource allocation procedure using the hybrid secure O-HMACSHA3 technique, which requires two parameters: T, representing the list of tasks, and R, representing the list of available resources. This process ensures security and optimized execution patterns for the given task lists. The first phase of O-HMACSHA3 is related to protection by generating secure keys for transmission between the client and server. This hybrid module enhances authentication for every call made to the server. The tasks with their corresponding information create a queue on the cloud server (T). The PSO optimization process in the proposed architecture provides calculated results for execution patterns and is connected to a resource parameter table to update resource status and allocations. The approved list of resources is associated with the given tasks, initiating their execution. The execution process records various metrics to track the effectiveness of the implemented method, as defined in pseudo code 1.

Input: List of tasks T, resource list R

Output: exe_tasks, unexecuted waiting_score

1. Listen to tasks queue T from the user end
2. Hybrid-HMAC authentication layer initialized
    - Generate secret key for secure transmission
    - Add authentication layer with generated secrets
3. Validate keys at cloud server for authorization
4. Analyze task t_i from tasks queue T
5. exe_tasks = 0, unexecuted =0
6. for each $t_i \in$ T do
7. if status($t_i \in$ {Init, pending}) then
8. h_id= allocate_pso(t_i) // h_id is host id from the available list
9. if waiting at resource==approved
10. location_res=where(h_id)
11. if location_res==matched
12. Allocate_task(t_i) →get_resource(h_id)
13. if resource reply==false
14. Process.kill(t_i)
15. end if
16. end if
17. exe_tasks= exe_tasks+1 // exe_tasks are successfully executed tasks list
18. else if status($t_i \in$ {crash, fail})
19. unexecuted= unexecuted+1 //unexecuted are the list of unexecuted tasks due to any reason
20. else
21. waiting_score= waiting_score+(ideal_en*time)
22. end if
23. end for

return exe_tasks, unexecuted, waiting_score

Pseudo Code 1 Secure Resource Allocation Process with O-HMACSHA

6. EXPERIMENT ANALYSIS

This section represents the analysis of investigational results attained from the implemented methods in comparison to existing methods. The implemented system evaluated the performance of the secure and optimized resource allocation approach in task scheduling. Various metrics related to task scheduling turnaround time and energy was considered in this research. The calculations were performed using MATLAB simulation language with a GUIDE interface. The simulations were conducted on machines equipped with an Intel Core i3 processor and 8 GB RAM. The structural details of the simulation analysis are provided in Table 10.

Table 10 Parameters of Simulation Tool

Metrics	Values
Simulation tool	MATLAB with GUIDE Version 2018a
Computing power	Intel i3 processor
Memory machine	8GB

This section describes the proposed O-HMACSHA3 and FSAOS methods, comparing their parameters such as EC and turnaround time to the existing method.

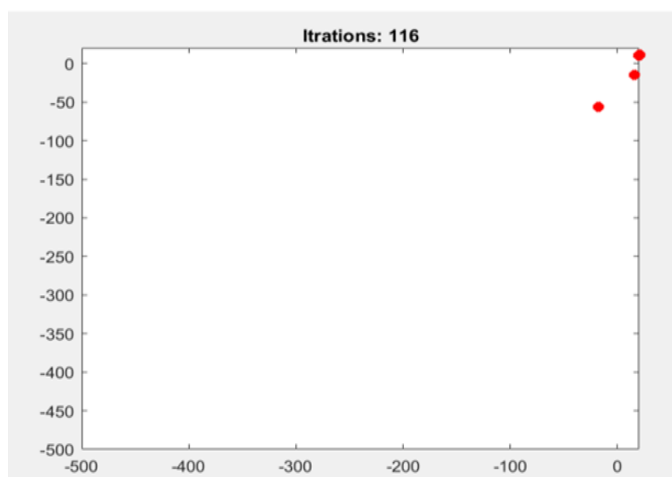

Figure 4 PSO Optimizer to Search for Best Cost Values

The task execution depends on the deployed solutions and their calculated cost, as presented in Figure 4. The main

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/221890 Volume 10, Issue 3, May – June (2023)

RESEARCH ARTICLE

motive is to minimize the cost function (CF), which processes the tasks and the resource list to determine allocation patterns and generate the cost of execution. This process helps reduce complexity and achieve a high accuracy rate.

```
processing tasks from the given list for location(re-allocate): 1
processing tasks from the given list for location(re-allocate): 2
processing tasks from the given list for location(re-allocate): 3
processing tasks from the given list for location(re-allocate): 4
processing tasks from the given list for location(re-allocate): 5
processing tasks from the given list for location(re-allocate): 6
processing tasks from the given list for location(re-allocate): 7
processing tasks from the given list for location(re-allocate): 8
processing tasks from the given list for location(re-allocate): 9
processing tasks from the given list for location(re-allocate): 10
processing tasks from the given list for location(re-allocate): 11
processing tasks from the given list for location(re-allocate): 12
```

Figure 5 Task Allocations in O-HMACSHA3

The task allocation process assigns resources to the input task queues. The list of tasks can be reallocated when a high-priority task emerges during execution. Figure 5 illustrates the reallocation process for the current queue in execution.

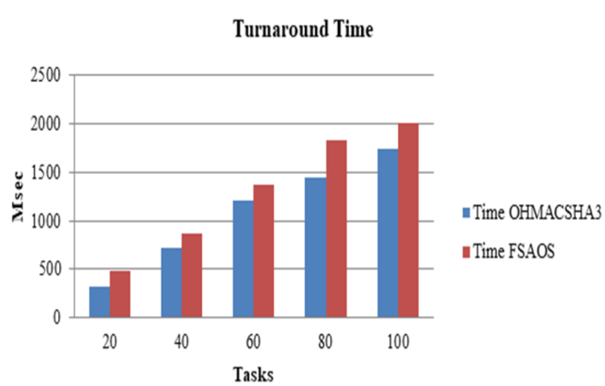

Figure 6 Comparative Analysis with Proposed and Existing Method: Turnaround Time (msec)

Turnaround time, depicted in Figure 6, represents the overall time for processing all the queues in a single execution. It indicates the time required for positive implementation of a procedure in the cloud environment. A shorter turnaround time signifies higher performance of the processing algorithms. The execution structure, along with the parameters and test cases, ensures high performance of the researched architecture for processing the queues.

Table 11 Comparison of turnaround time

Number of tasks queues	Time O-HMACSHA3(millisec)	Time FSAOS(millisec)[19]
20	316	479
40	721.6	865.2
60	1215.2	1378.8
80	1447	1838
100	1745	2014

The proposed approach undergoes a validation process involving the execution of various queues to evaluate performance metrics. Table 11 presents the performance comparison between the recently developed existing model and the planned method. The execution results demonstrate the superiority of the O-HMACSHA3 method in all cases.

Table 12 Comparison of Energy Consumption

Number of tasks	Energy O-HMACSHA3(Joule)	Energy FSAOS(Joule)[19]
20	47.7	57.3
40	197.9	219.8
60	414.2	504.8
80	689.2	831.4
100	752.1	924.2

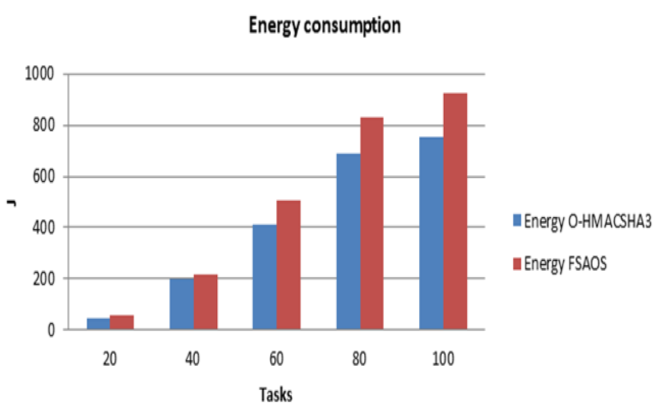

Figure 7 Comparative Analysis with Proposed and Existing Method - Energy Consumption j (joule)

Table 12 and Figure 7 define energy consumption as the cost of executing tasks. Lower consumption is desired for efficient execution. The variation in implementation aims to validate the load and performance under different loads. The energy consumption in the planned environment is comparatively lower, indicating the high performance of the introduced approach. The different cases, ranging from small to heavy loads on the allocation side, demonstrate variations in energy consumption. As the system works more, energy consumption tends to increase. However, compared to the existing approach, the planned approach shows lower consumption in all cases.

7. CONCLUSION AND FUTURE WORK

The research analysis has developed and introduced a secure and optimized resource allocation method for TS. The method has been compared with O-HMACSHA3 and FSAOS [19] methods and the obtained results confirm improvements in turnaround time and energy consumption. The implemented method utilizes the PSO method for optimization and

RESEARCH ARTICLE

enhancement of solutions. The proposed O-HMACSHA3 method is presented as an enhancement for reliable resource scheduling access in the cloud environment. The research aims to address existing issues by developing soft security measures for reliable resource scheduling access using PSO optimization in the cloud environment. The performance of the proposed cloud resource scheduling technique is calculated and compared with other task scheduling methods, namely PSO and FSAOS [19], in terms of time and EC. The research method resolves existing problems and improves performance metrics, with energy consumption at 47.7 joules and TAT (Turnaround Time) at 316 msec, compared to the proposed method, which achieved 57.3 joules and 479 msec for 20 tasks, surpassing other existing methods.

As for future work, the research method will focus on enhancing Quality of Service (QoS). The attained consequences confirm the effectiveness of the implemented O-HMACSHA3 process. Furthermore, a SHA3-512 strategy will be implemented to ensure reliable service delivery to clients in the cloud environment. Energy-conscious plans will also be discussed, which have garnered significant interest from the research community. A work schedule considering energy consumption will be created and presented in an actual cloud environment.

REFERENCES

[1] Gupta, A., Garg, R. “Load balancing based task scheduling with ACO in cloud computing”, In 2017 International Conference on Computer and Applications (ICCA), 2017, (pp. 174-179).IEEE.

[2] Rathore, N., Chana, I. “ Load balancing and job migration techniques in grid: a survey of recent trends”, Wireless personal communications, 2014, 79(3), 2089-2125.

[3] Grewal, S. K., Mangla, N. “Deadline based Energy Efficient Scheduling Algorithm in Cloud Computing Environment”, In 2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT),2021, (pp. 383-388). IEEE.

[4] Ms.Saranya G, Mr.Srinivasan J “Load Balancing Algorithm In Task Scheduling Process Using Cloud Computing”, International Research Journal of Engineering and Technology (IRJET), 2016, vol 3(5), pp. 1-8.

[5] Singh S, Chana I “Cloud resource provisioning: survey, status and future research directions”. KnowlInfSyst, 2016, 49(3):1005–1069

[6] Panda, SK., and Prasanta K. J. "Efficient task scheduling algorithms for heterogeneous multi-cloud environment." The Journal of Supercomputing 71 (2015): 1505-1533.

[7] Liu L, Mei H, Xie B “Towards a multi-QoS human-centric cloud computing load balance resource allocation method”, J Supercomput 2016, 72(7):2488–2501.

[8] Panda, S. K., Jana, P. K. “Load balanced task scheduling for cloud computing: A probabilistic approach”, Knowledge and Information Systems, 2019, 61(3), 1607-1631.

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

[10] Panda, S. K., Jana, P. K. “ A multi-objective task scheduling algorithm for heterogeneous multi-cloud environment”,In 2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV) 2015 (pp. 82-87).IEEE.

[11] Zuo, L., Shu, L., Dong, S., Zhu, C., Hara, T. “A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing”, Ieee Access, 2015, 3, 2687-2699.

[12] Shishido, H. Y., Estrella, J. C., Toledo, C. F. M., Arantes, M. S. “Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds”. Computers & Electrical Engineering, 2018, 69, 378-394.

[13] Mubeen, A., Ibrahim, M., Bibi, N., Baz, M., Hamam, H., Cheikhrouhou, O. “Alts: An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing”. Processes, 2021,9(9), 1514.

[14] Potluri, S., Rao, K. S. “Optimization model for QoS based task scheduling in cloud computing environment”. Indonesian Journal of Electrical Engineering and Computer Science, 2020, 18(2), 1081-1088.

[15] Singh, H., Bhasin, A., Kaveri, P. R. “QRAS: efficient resource allocation for task scheduling in cloud computing”, SN Applied Sciences, 2021, 3(4), 1-7.

[16] Ibrahim, M., Nabi, S., Hussain, R., Raza, M. S., Imran, M., Kazmi, S. A., Hussain, F. “A comparative analysis of task scheduling approaches in cloud computing”, In 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), 2020, (pp. 681-684).IEEE.

[17] Afzal, S., Kavitha, G. “Load balancing in cloud computing–A hierarchical taxonomical classification”, Journal of Cloud Computing, 2019, 8(1), 1-24.

[18] Kumar, M., Dubey, K., Sharma, S. C. “Elastic and flexible deadline constraint load balancing algorithm for cloud computing”. Procedia Computer Science, 2018, 125, 717-724.

[19] Gabi, D., Dankolo, N. M., Muslim, A. A., Abraham, A., Joda, M. U., Zainal, A., Zakaria, Z, ”Dynamic scheduling of heterogeneous resources across mobile edge-cloud continuum using fruit fly-based simulated annealing optimization scheme”, Neural Computing and Applications, . 2022, pp. 1-21.

[20] Saleh, H., Nashaat, H., Saber, W., Harb, H. M. “IPSO task scheduling algorithm for large scale data in cloud computing environment”, IEEE Access, 2018, 7, pp. 5412-5420.

[21] Farhadian, F., Kashani, M. M. R., Rezazadeh, J., Farahbakhsh, R., & Sandrasegaran, K. “WITHDRAWN: An efficient IoT cloud energy consumption based on genetic algorithm”, Digital Communications and Networks, 2019.

[22] Dordaie, N., Navimipour, N. J. “A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments”, ICT Express, 2018, 4(4), 199-202.

[23] Selvakumar, A., Gunasekaran, G. “A novel approach of load balancing and task scheduling using ant colony optimization algorithm”, International Journal of Software Innovation (IJSI), 2019, 7(2), 9-20.

[24] Torabi, S., Safi-Esfahani, F. “A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing”, The Journal of Supercomputing, 2018, 74(6), 2581-2626.

[25] Basu, S., Kannayaram, G., Ramasubbareddy, S., Venkatasubbaiah, C. “Improved genetic algorithm for monitoring of virtual machines in cloud environment”. In Smart Intelligent Computing and Applications 2019, (pp. 319-326). Springer, Singapore.

[26] Basu, S., Karuppiah, M., Selvakumar, K., Li, K. C., Islam, S. H., Hassan, M. M., & Bhuiyan, M. Z. A. “ An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment”, Future Generation Computer Systems, 2018, 88, 254-261.

[27] Shukri, S. E., Al-Sayyed, R., Hudaib, A., Mirjalili, S. “Enhanced multi-verse optimizer for task scheduling in cloud computing environments”, Expert Systems with Applications, 2021, 168, 114230.

[28] Velliangiri, S., Karthikeyan, P., Xavier, V. A., Baswaraj, D. “Hybrid electro search with genetic algorithm for task scheduling in cloud computing”, Ain Shams Engineering Journal, 2021, 12(1), 631-639.

RESEARCH ARTICLE

[29] Negi, S., Rauthan, M. M. S., Vaisla, K. S., Panwar, N. " CMODLB: an efficient load balancing approach in cloud computing environment". The Journal of Supercomputing, 2021, pp. 1-53.
[30] Krishna, A. V., Ramasubbareddy, S., Govinda, K. " Task scheduling based on hybrid algorithm for cloud computing", In International Conference on Intelligent Computing and Smart Communication 2019 (pp. 415-421).Springer, Singapore.
[31] Topcuoglu, H., Hariri, S., Wu, M. Y. "Performance-effective and low-complexity task scheduling for heterogeneous computing", IEEE transactions on parallel and distributed systems, 2002, 13(3), 260-274.
[32] Mao, Y., Chen, X., & Li, X. (2014). Max–min task scheduling algorithm for load balance in cloud computing. In Proceedings of International Conference on Computer Science and Information Technology (pp. 457-465).Springer, New Delhi.
[33] Li, C., Jiang, W., & Zou, X. (2009, December). Botnet: Survey and case study. In 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC) (pp. 1184-1187). IEEE.
[34] R. K. Mondal, E. Nandi and D. Sarddar, "Load balancing scheduling with shortest load first," International Journal of Grid Distribution Computing, vol. 8, no. 4, pp. 171-178, Oct. 2015
[35] Mohapatra, S., Mohanty, S., &Rekha, K. S. (2013). Analysis of different variants in round robin algorithms for load balancing in cloud computing. International Journal of Computer Applications, 69(22), 17-21.
[36] Devi, D. C., & Uthariaraj, V. R. (2016). Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks. The scientific world journal, 2016.
[37] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.
[38] Mahato, D. P., Singh, R. S., Tripathi, A. K., &Maurya, A. K. (2017). On scheduling transactions in a grid processing system considering load through ant colony optimization. Applied Soft Computing, 61, 875-891.
[39] Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A genetic algorithm (ga) based load balancing strategy for cloud computing. Procedia Technology, 10, 340-347.
[40] LD, D. B., & Krishna, P. V. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied soft computing, 13(5), 2292-2303.
[41] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.
[42] Navimipour, N. J., &Milani, F. S. (2015). Task scheduling in the cloud computing based on the cuckoo search algorithm. International Journal of Modeling and Optimization, 5(1), 44.
[43] Gupta, G., & Mangla, N. (2021). Energy Aware Metaheuristic Approaches to Virtual Machine Migration in Cloud Computing. In IOP Conference Series: Materials Science and Engineering (Vol. 1022, No. 1, p. 012050). IOP Publishing.
[44] Almezeini, N., & Hafez, A. (2017). Task scheduling in cloud computing using lion optimization algorithm. International Journal of Advanced Computer Science and Applications, 8(11).
[45] Kumar, M., Sharma, S. C., Goel, A., & Singh, S. P. (2019). A comprehensive survey for scheduling techniques in cloud computing. Journal of Network and Computer Applications, 143, 1-33.
[46] Dubey, K., Kumar, M., & Sharma, S. C. (2018). Modified HEFT algorithm for task scheduling in cloud environment. Procedia Computer Science, 125, 725-732.
[47] Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013, February). User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In 2013 National Conference on Parallel computing technologies (PARCOMPTECH) (pp. 1-8).IEEE.
[48] Mao, Y., Chen, X., & Li, X. (2014). Max–min task scheduling algorithm for load balance in cloud computing. In Proceedings of International Conference on Computer Science and Information Technology (pp. 457-465).Springer, New Delhi.
[49] Mondal, R. K., Nandi, E., &Sarddar, D. (2015). Load balancing scheduling with shortest load first. International Journal of Grid and Distributed Computing, 8(4), 171-178.
[50] Ramezani, F., Lu, J., & Hussain, F. K. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. International journal of parallel programming, 42(5), 739-754.
[51] Sun, W., Zhang, N., Wang, H., Yin, W., &Qiu, T. (2013, December). PACO: A period ACO based scheduling algorithm in cloud computing. In 2013 International Conference on Cloud Computing and Big Data (pp. 482-486).IEEE.
[52] Chhabra, A., Singh, G., & Kahlon, K. S. (2020). QoS-aware energy-efficient task scheduling on HPC cloud infrastructures using swarm-intelligence meta-heuristics. Comp. Mater. Cont, 64, 813-834.
[53] Mangla, N. (2021, October). Soft Security Resource Scheduling Issues in Cloud Computing: A Review. In 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC) (pp. 678-684). IEEE.

Authors

Dr. Himanshu, working as an Assistant Professor in Computer Science & Engineering Department at Chandigarh University, Mohali, Punjab. She has 8 Years of teaching experience and 3Years of research experience. She wrote 10 research papers in various journals and conferences. Her area of interest includes Cloud computing, Block Chain, Image Processing, Distributed database, Data mining etc.

Dr. Neeraj Mangla, working as a professor in Computer Science & Engineering Department at Maharishi Markandeshware Engineering College, Maharishi Markandeshware (Deemaed to be University) Mullana, Ambala, Haryana. He has 20 Years of teaching experience and 12 Years of research experience. He wrote 50 research papers in various journals and conferences. His area of interest includes Cloud computing, Distributed database, Data mining etc.

How to cite this article:

Himanshu, Neeraj Mangla, "A Hybrid Secure and Optimized Execution Pattern Analysis Based O-HMACSHA 3 Resource Allocation in Cloud Environment", International Journal of Computer Networks and Applications (IJCNA), 10(3), PP: 359-370, 2023, DOI: 10.22247/ijcna/2023/221890.