International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

Latency-Aware Task Scheduling Based on Hybrid Meta-Heuristic Optimization Algorithm in Fog Computing

Ali Garba Jakwa, AbdulsalamYa,u Gital, Souley Boukari and Fatima Umar Zambuk

Department of Mathematical Sciences, Faculty of Science
AbubakarTafawa Balewa University

Date of Submission: 05-03-2023 Date of Acceptance: 15-03-2023

ABSTRACT
Task scheduling in fog computing is one of the areas where researchers are having challenges as the demand grows for the use of Internet of Things (IoT) to access cloud computing resources. Many resource-scheduling and optimization algorithms were used by many researchers in fog computing; some used single techniques while others used combined schemes to achieve dynamic scheduling in fog computing. This paper proposedHybrid Meta-Heuristics Optimization Algorithm (HMOA) for latency-aware task scheduling in fog computing, the study combined Particle Swarm Optimization (PSO) Meta-heuristics and deterministic Spanning Tree (SPT) to achieve task scheduling with the intention of eliminating the drawbacks of the two algorithms when used separately, the PSO was used to schedule user task requests among fog devices, while hybrid MPSO-SPT was used to perform resource allocation and resource management in the fog computing environment. The study implemented the algorithm using iFogSim in which the performance of the algorithm was evaluated, assessed and compared with other state of art task scheduling and resource management algorithms. The proposed method performed better in terms of execution time with 5ms in relation to best case average case and worst case scenarios, load balancing, cost of execution with $544100and communication latency with 3.45ms. The results were compared with MPSO-MCSO, FCFS, SJF and NBIHA algorithms, and the study recommends that further research should carried out to incorporate mobility in fog as mobility is becoming one of the major trends in research and cloud computing.

KEYWORDS: Meta-Heuristic Algorithms, Task Scheduling, Execution Time and Fog Computing

I. INTRODUCTION

The internet has been most of the popular and largest mode of communication among people and government all over the world, as such a vast number of devices are connected and consuming as well as generating data by offering variety of computing services with different set of tasks. Such devices can be fixed or mobile, this days, many applications and services running on fixed or mobile devices mostly depend on remote services as in cloud computing servers to store data and perform data processing, scheduling task in such environment requires a dynamic approach to have equal distribution of task among the fog nodes.

Task scheduling and load balancing is one of the most important segment in fog computing infrastructure, the user tasks are of different patterns from different sources, some task are fixed in nature and some are mobile in nature.

Task scheduling in fog computing is one of the areas where researchers are having challenges as the demand grows for the use of Internet of Things (IoT) to access cloud computing resources. Many resource-scheduling and optimization algorithms were used by many researchers in fog computing; some used single techniques while others used combined schemes to achieve dynamic scheduling in fog computing, many optimization techniques are reassessed based on deterministic or meta-heuristics to find out solution to scheduling problem in fog computing, but their methods could not produce efficient task scheduling in fog computing due to slower convergence rate.

Many researchers[1]have conducted a research on task scheduling in fog computing by using different meta-heuristics algorithms like Modified Particle Swarm Optimization (MPSO), Modified Cat Swarm Optimization (MCSO), that

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

are heuristics to achieve optimal resource utilization and resource management, most of the research observed;[1], presented their results based on energy consumption, execution cost in fog nodes by using these heuristic algorithms which always depend on exploration and exploitation, therefore they tend to be slow and higher convergence rate when dealing with large search space, to overcome this drawbacks, this research intended to consider introducing a Spanning Tree (SPT) algorithm which is a high speed deterministic optimization algorithm in order to improve the search space.

This paper proposed a latency-aware task scheduling algorithm with efficient execution time in fog computing environment.

The paper is organized based on the following sections; section I presented the introduction of the paper which includes general terms and background about fog computing, section II presented the layers of fog computing which includes the general architecture of fog computing, section III presented the literatures that are related to fog computing with regards to task scheduling, section IV presented the methodology of the proposed system design which include the system architecture and the various parameters required for the development of the proposed system and finally section V presented the conclusion and future directions of the research.

The Architecture of Fog Computing

Fog Computing is one of the Cloud computing layer that provides access and seamless transmission of data to the end users with faster and efficient communication between the cloud and the end users task scheduling is one of the important area of research that researchers are exploring efforts and talents in other to device a reliable and efficient scheduling method in fog and cloud computing in general.

The fog computing architecture is divided into three major layers which includes; the cloud layer, the fog layer and the edge layer in which processing are carried out hierarchically as shown the figure 1 below;

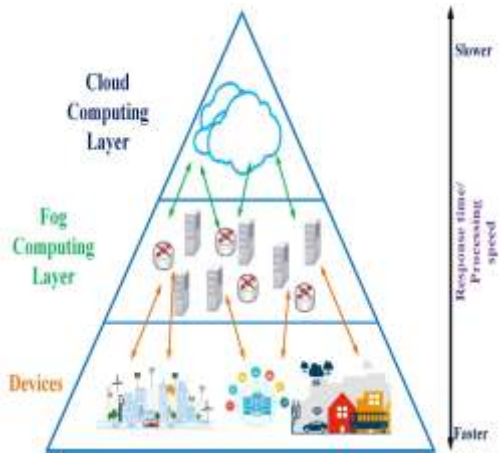

Figure 1: Fog Computing Architecture

Critical Review of Literature

Data Placement in Fog Computing; is the processing of placing and scheduling resources among the n number of fog devices in order to reduce the traffic on a particular fog node, such that optimal resources are shared among the available fog nodes in the computing environment, data placement can be achieved by employing different placement algorithms which include Mapping algorithm, Cloud-only, Edge-ward, etc.[2].

It has not been the intention that fog computing should replace cloud computing, but rather to be used in conjunction with more central cloud units to reduce delay, provide fast computing and reduce the cost of processing.

Fog nodes can be classified into two types: resource-poor devices such as routers, set-top-units, wireless access points (WAP); and resource-rich machines such as cloudlets, that can be considered a small scale version of cloud data centers, giving massive processing to mobile devices with low latency[1].

Pradeep & Jacob, (2018) proposed a hybridization of cuckoo search and gravitational search algorithm (CGSA) for task scheduling. The purpose of the design was to exploit the merits of both cuckoo search (CS) and gravitational search algorithms (GSA) while avoiding their drawbacks. The performance of the algorithm was analyzed based on the different evaluation measures. The algorithms like GSA, CS, Particle swarm optimization (PSO), and genetic algorithm (GA) were used as a comparative analysis, the proposed model outperform the other methods based on the results presented. The algorithms used are both heuristic in nature therefore they tend to be slow thereby leading to higher execution time.

[1]Proposed a nature inspired task scheduling algorithm in fog computing, they

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

modeled and proposed a novel bio-inspired hybrid algorithm (NBIHA) which is a hybrid of modified particle swarm optimization (MPSO) and modified cat swarm optimization (MCSO). In the proposed scheme, the MPSO was used to schedule the tasks among fog devices and the hybrid of the MPSO and MCSO was used to manage resources at the fog device level. In the proposed approach, the resources were assigned and managed on the basis of the demand of incoming requests. The main objective of their proposed work was to reduce the average response time and to optimize resource utilization by efficiently scheduling the tasks and managing the fog resources available, their proposed model outperformed other related research in terms of optimal resource utilization and energy saving.

But combining two heuristics algorithms which perform their search through exploitation and exploration to achieve a solution is cumbersome since the algorithms are having slowerconvergence rate compare to other none-heuristic algorithms.

Spanning Tree (SPT) algorithm have been used by many researchers in improving the convergence rate of heuristic algorithms as observed in[3] in their research, they proposed ant colony optimization population based and deterministic spanning tree ACO-SPT for task scheduling in cloud computing, their proposed algorithm achieved a faster convergence with loop free, with least make span time based on their simulated results, and their result outperforms other algorithms in terms of load balancing and proposed future work in assessing the algorithm performance in multi-tier environment and software Defined Network (SDN). However their work is limited to resource management in cloud without extending fog or edge computing.

II. METHODOLOGY

Thispaper proposed a Hybrid Meta-heuristics Optimization algorithm (HMOA) for task scheduling in fog computing environment, with the goal to design latency-aware fog resource services by optimizing Modified Particle Swarm Optimization (MPSO) using one of the optimization algorithms which is Modified Cat Swarm Optimization (MCSO) algorithm and injecting Spanning Tree (SPT) deterministic Algorithm in order to improve the convergence rate with reduced latency in fog computing.

Proposed Method

The proposed method is the combination of Meta-heuristic and deterministic spanning tree algorithms presented as follows;

Hybrid Meta-heuristics Optimization Algorithm (HMOA)

The main purpose of this proposed scheduling algorithm is to find optimal schedules for executing task in the fog environment. The main idea of this proposed algorithm is to use Modified Cat Swarm Optimization (MCSO) algorithm and Modified Particle Swarm Optimization (MPSO) to solve the problem of task scheduling by injecting Spanning Tree (SPT) Algorithm in order to improve the convergence speed of exploration and exploitation in the search space.

The optimization criteria in the proposed algorithm is to address the shortcomings observed by the MPSO and MCSO algorithms when used, as both of them are heuristic algorithms and they have a problem of crossover when handling large search space, therefore this algorithm is the combination of Hybrid MPSO and SPT to achieve resource management with latency in mind in fog computing and subsequent sending the tasks to cloud for further processing[4].

Meta-Heuristic algorithms

The concept of Meta-heuristics algorithms are set of problem solving methods in which are desinged to find, select, or generate a heuristic that can produce a significantly better result to an optimization problem with few iterations.

This algorithm provide better result through exploration and exploitation specifically with very limited computational effort or non complete information which is applicable to wide number of problems for task scheduling in fog computing.

The following are examples of meta-heuristics algorithms; Particle Swarm Optimization (PSO), Cuckoo Search (CS), Cat Swarm Optimization (CSO), Genetic algorithm (GA) and so on. These algorithms have been used and reasonably performed better in optimizing task scheduling and cloud service providers throughput in the area of minimizing makespan, balancing load and scalability by providing shortest optimal results within shortest period[1].

Deterministic Spanning Tree (SPT) Algorithm

Spanning Tree (SPT) Algorithm is a member of deterministic algorithms which are used for finding an optimal path in a search space

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

through graph means of visiting nodes and therefore the Spanning Tree is inform of graph and its subset which is having all the nodes connected with possible number of arcs. Generally, the spanning tree has no loops or cycles and cannot be disconnected[3].

However it is good in handling problems that require alternative routes in decision making especially task scheduling. The loops in the transition operators that may cause relative influence will be eliminated by the Improved SPT in the set $G = (N, M)$.

The function of the Spanning Tree here is to stop all redundant paths in cloud user request for job that may cause loop allowing convergence delay, this will allow existence of single logical path between all destinations in the fog devices. In the event where users request is intentionally denied from leaving or entering a path, the path will be termed as blocked path.

Convergence factor

Consider;

$$\lim_{t\to\infty} \text{Makespan (t)} = \lim_{t\to\infty} \sum_{j}^{m} \text{Execution time task } (T_i) \quad \ldots \quad \text{equation (1)}$$

In order to obtain faster convergence rate, the limit for the make span time must be taking to cover the task execution time length.

Problem formulation

The aim of this proposed system is to design latency aware task scheduling system in fog computing environment which is focused on load balancing and resource management. In this system, tasks are being described as methods that define user's service demand which can be in the form of mobile user, web user, or internet users. User tasks or requests $t_u\{t_1, t_2, t_3, t_4, \ldots\ldots, t_n\}$ and available fog devices $fd_j\ \{fd_1, fd_2, fd_3, fd_4, \ldots\ldots fd_k\}$ are used to schedule these tasks and cloud resources. Our proposed system used MPSO and SPT for task scheduling and load balancing by using best fit cloud and fog devices for request processing, when tasks are being scheduled, the average response time of the fog devices was found based on the given equation;

$$\text{Average Response Time} = t_2\left[\sum_{x=1}^{k} FD(x)\right] - t_1\left[\sum_{x=1}^{k} FD(x)\right] \quad \ldots \quad \text{equation (2)}$$

Where FD is the fog devices, x is the user task from 1 to k and t_1 , t_2 , is the initial and final time.

Best fit of the fog device can be obtained by finding the fitness value based on the given equation 5 and the equation can be used to calculate the resource demand of a given task.

$$\text{Fitness Value} = \frac{(R)\ \sum_{FD=0}^{k} FD(j)}{\sum_{x=1}^{n} (RD)} \quad \ldots \quad \text{equation (3)}$$

Where RD is the resource demand, R is the resources, FD is the fog devices; x is the number of task from 1 to n.

Experimental Setup

There are many simulators that are available for evaluating fog computing research and assess the performance, but this research experiment used iFogSim simulator as a platform for simulating the proposed model. iFogSim is high performance toolkit to model and simulate the networks of Edge Computing, Internet of Things and Fog Computing. iFogSim integrates the resource management techniques which can be further customized as per the research area. The simulation with iFogSim works in association with the CloudSim. CloudSim is a widely used library for the simulation of cloud based environment and resource management. The layer of CloudSim exists to handle the events between the components of Fog Computing using iFogSim [2]

Performance evaluation

The evaluation metrics for this paper is based on execution time, load balancing, processing cost and communication latency.

System Architecture

The following subsections presented the components used in the proposed system as presented thus;

Cache based load balancing

The load balancing mechanism in task scheduling is the ability of the system to distribute user tasks equally among the fog devices with optimal utilization of equal tasks all the time. The idea of cache based load balancing, the tasks that are frequently sent to system are being kept in

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

storage location in which requested task can subsequently be accessed easily without going back to the main task pool and reprocess again, and this scenario can improve the processing of the algorithms whenever used in task scheduling.

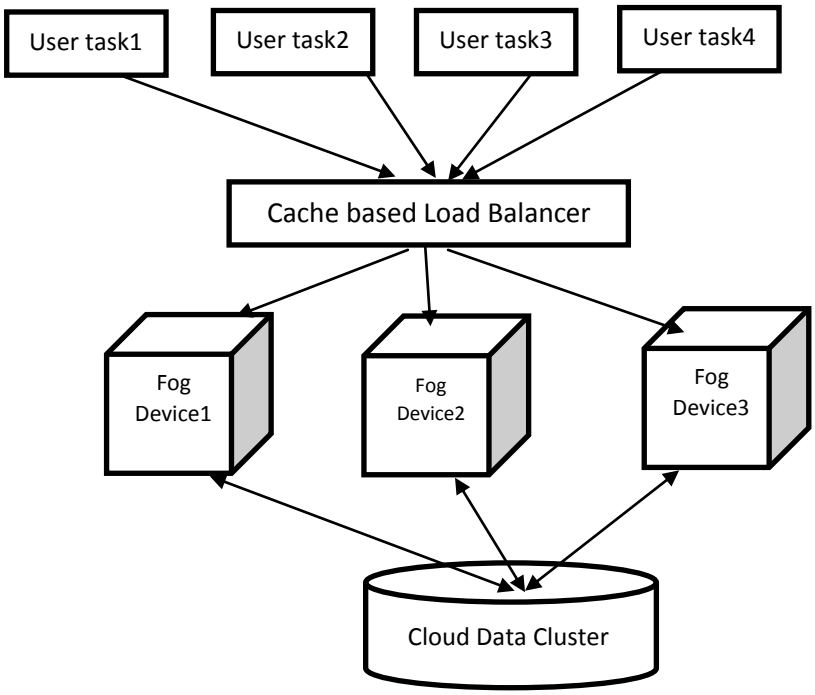

Figure 2: Load balancing workflow

Latency-aware task scheduling

In this regard, we consider non-deterministic and spanning tree latency-aware task scheduling in fog computing in other to determine the run-time requirement adjustment of task scheduling, this method is focused on module placement which is intended to forwarding task toward inactive resources, this method was adopted from[3] by injecting spanning tree algorithm; which employed similar method in determining required time for resource management in fog computing.

This proposed method is a form of heuristic and deterministic method in which the method is targeted at achieving better latency among communication devices, in this method, the incoming task will be forwarded to free nodes immediately through inspection of switch allocation[3]

Latency in fog computing

Application performance can be achievd by reducing the latency betweeen fog devices and edges through proper use of placement algorithm, the approach used the method presented in the equation (1), the symbol $\propto$ is the delay incured while capturing data in form of turples, μ is the time required to perform motion detection and upload, while θ is the time required to display the detected task to the required user interface[9].

$$\text{Latency} = \propto + \mu + \theta \quad \ldots \quad \text{equation (1)}$$

Presentation of results

The previous section presented the general overview of the concept of cloud and fog computing,followed by different literatures and techniques used for task scheduling in fog environment, related works was also observed by different authors on the how tasks are being scheduledusing different algorithms in fog computing and their subsequent implementation; algorithms like heuristic and deterministic algorithms were observed both in hybrid and non-hybrid formats, in which this research adopted some of the techniques where heuristic and deterministic algorithms are proposed for task scheduling in fog computing, based on observations of different researchers on the context of using either heuristic or deterministic separately.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

The simulation results are presented in the following sections, by considering the following evaluation metrics; Execution time, Cost of execution, load balancing and communication latency as presented further.

III. RESULTS

The Hybrid Meta-Heuristic Optimization Algorithm (HMOA) was used in this research for task scheduling and resource allocation. The proposed framework is used to obtain the best fit fog device for processing of all incoming tasks, since the derivation of original Modified Particle Optimization (MPSO) Algorithm, our new framework was slightly modified by introducing Spanning Tree Algorithm to the MPSO having a hybrid algorithm for solving problem of task scheduling and resource allocation in fog computing environment, we also look at the communication latency between fog devices.

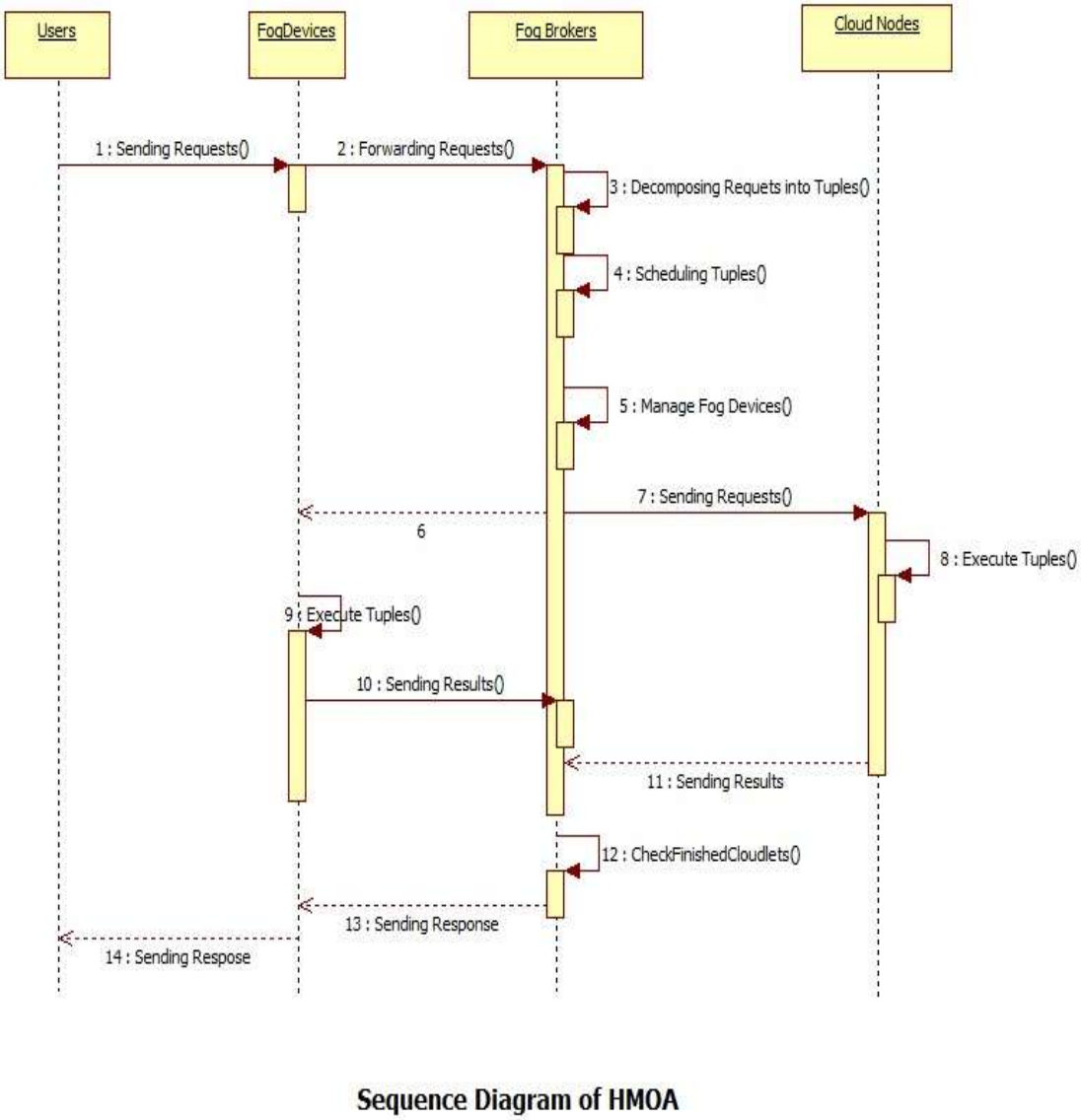

Figure 3: Sequence Diagram depicting HMOA

Figure 3 above depict the operation of Hybrid Meta-Heuristic Optimization Algorithm (HMOA)

In fog-cloud collaborationusing the iFogSim toolkit, user task request are sent which are received by the fog devices. Fog broker manages the request generated by the users as well as the fog devices, the request are then divided into tuples and scheduled tasks are then managed by the fog devices based on the HMOA.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

Tuples are then sent to the cloud devices and fog devices as shown in the sequence diagram after the complete process, the fog brokers check the tasks completion, the results are then sent back to the user through the fog device after compilation.

Performance Evaluation

Simulations have been conducted extensively using iFogSim in a cloud-fog environment in order to assess the performance of the proposed approach which is HMOA. There have been varied processing and usage cost in the cloud and fog nodes, in this work, we assumed that each fog node has its own processing limits (measured by MIPS – million instructions per second), including CPU, memory, and transmission capacity utilization cost. In our experimental setup, the Fog framework consist of 15 processing nodes; 5 cloud and 10 fog node as specified in table 1 below, which will be varied consecutively as required in the simulation process and experimental procedure and parameter required.

Table 1: Evaluation parameters

Parameters	**Fog**	**Cloud**
Number of Nodes	10	5
CPU MIPS	[500, 2000]	[3000 – 10000]
CPU usage cost		

Source: (Experiment 2023)

Fog nodes have restricted processing power in the fog layer, for example, doors, switches, workstations or PCs and tablets. The mechanism that are in charge of taking care of requests, are servers or virtual machines in elite server farms in the cloud layer, in this regard, the preparation rate of cloud nodes is much quicker than fog nodes. Consequently, the expense of utilizing resources in the fog is cheaper than in the cloud as the distance between the user's tasks and the fog devices are closer than the cloud.

Fog framework is generally responsible for execution of all incoming requests from the client's sides. Each incoming request is divided into number of task tuples, which subsequently decomposed and evaluated upon the processing that they required.

The results for our proposed method were analyzed based on the following parts; execution time,cost of execution, cache based load balancing and communication latency. The proposed framework was also compared with other state of art scheduling algorithms in fog computing such as Modified Particle swarm optimization (MPSO) and Nature Bio-Inspired Hybrid Algorithm (NBIHA) etc.

Resource allocation by using task scheduling

As indicated in the research objectives this research is divided into two sections which includes latency-aware resource management and cache-based load balancing. To achieve part of the objectives, the proposed method used Hybrid Meta-Heuristic Optimization Algorithm (HMOA), the framework wishes to address task scheduling by combining Spanning Tree and MPSO algorithms, and presented the results as follows;

i. **Execution Time Analysis**

Execution time for this experiment was being categorized based on best, average and worst case results, different combinations of tasks and fog devices were explored with other algorithms to test the scheduling capabilities in terms of execution time, our proposed method presented better result in terms of execution time compared to other scheduling algorithms as shown in Figure. 4 below, this is because of the inclusion of the deterministic algorithm in the processing module.

Table2: Algorithm analysis based on execution time

S/N	Algorithm analysis	HMOA	NBIHA	FCFS	SJF
		Time (ms)			
1	Best Case	5	6	7	8
2	Average Case	10	15	15	11
3	Worst Case	15	17	18	19

Source: (Experiment 2023)

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

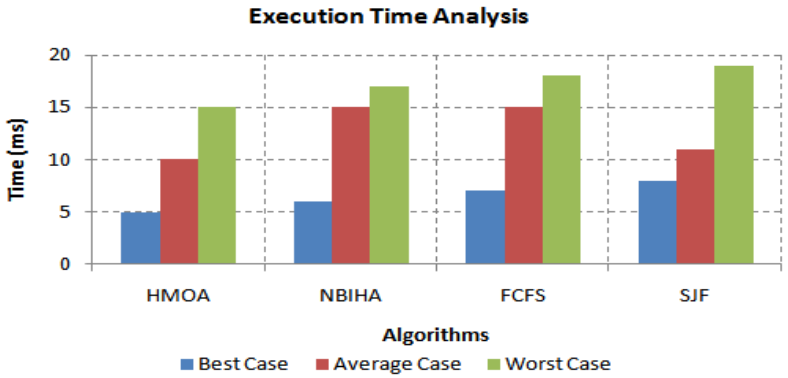

Figure 4: Execution Time Analysis

ii. **Processing Cost Analysis**

In the task scheduling and management, the hybrid and deterministic method reduces waiting time for the resources thereby reducing the cost of processing due faster convergence rate in which the execution time tend to be faster thereby reducing the processing cost among the fog and the edge devices as shown in Figure. 5. The value depicted in the table, are not real values of money,they are just numeric figures used for experimental process to differentiate between execution costs among the algorithms.

Table 3: Cost of Execution

Algorithms	HMOA	NBIHA	FCFS	SJF
Cost ($)	544100	544300	544600	545555

Source: (Experiment 2023)

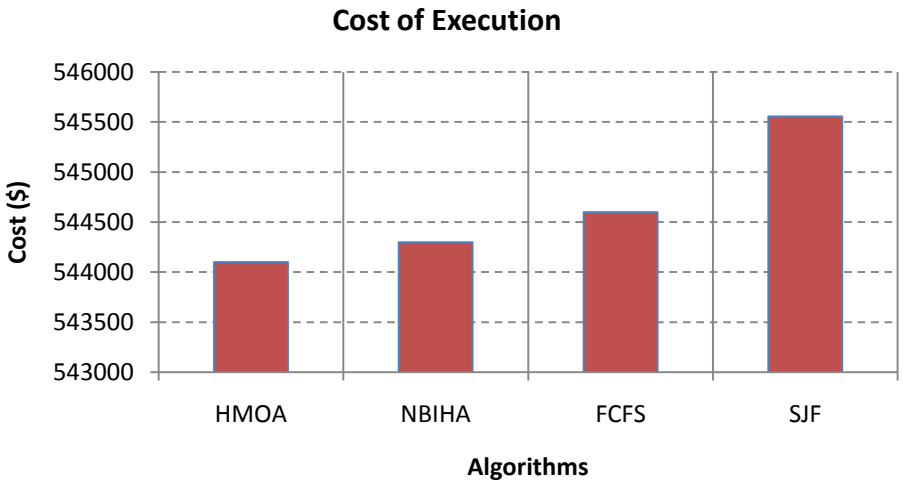

Figure 5. Cost of Execution of based on the algorithms

Load Balancing

Figure 6 below depicts the simulation results of the algorithm in which task are distributed equally among the fog devices as observed against each fog device ID.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

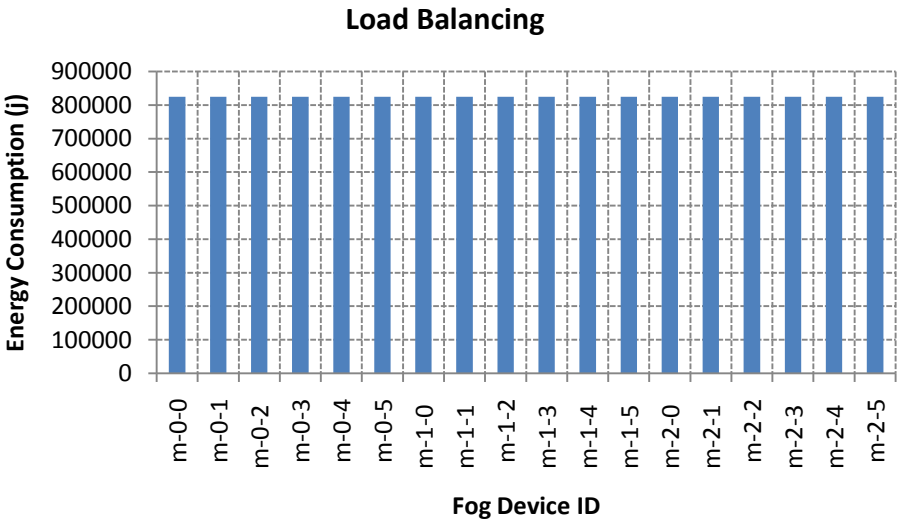

Figure 6. Load balancing

Latency in fog computing

The experimental result for the simulation is displayed in table 4 below and compared with state of the art latency analysis algorithms;

Table 4: Latency analysis

Number of Fog Devices	LJAYA Optimization	Latency-Aware Module Management	Heuristic and Deterministic Latency-aware Computing
5	1.1	2.2	1.0
10	2.16	4.3	2.01
15	2.89	7.015	2.56
20	3.2	9.3	3.1
25	3.43	19.9	3.33
30	3.65	34.2	3.45

Source: (Experiment 2023)

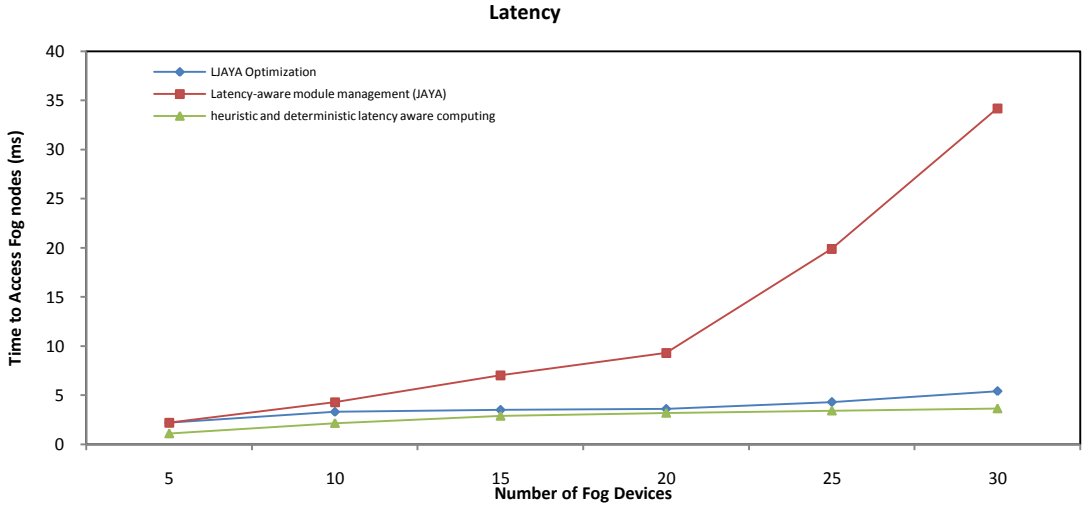

Figure 7.Fog devices against time

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 3 March 2023, pp: 786-798 www.ijaem.net ISSN: 2395-5252

Latency analysis

Based on the simulation result, the communication latency using heuristic and deterministic has no big difference with the response time, with this regard the study observed that communication latency and network response works hand in hand, our approach was also compared with other latency analysis algorithms and our model presented better access time between fog nodes. **Discussion**

In effort to achieve the stated objectives of the research, the study was evaluated and analyzed the proposed method using the following parameters: execution time (ms), Cost of Execution, Cache based load balancing and communication Latency.

The proposed method was comparedwith benchmark algorithms for task scheduling such as; NBIHA, MPSO, SJF and FCFS.We have also compared our proposed approach with HFSGA (Hybrid Flamingo Search with a Genetic Algorithm) meta-heuristic and deterministic approach in the case of resource management and allocation to fog nodes.

The results from the simulations showed that our proposed method has better results in the case of Execution time 5ms, Cost of Execution, Load balancing 17kj and communication Latency 3.45ms when compared to other benchmark state-of-the-art algorithms. In the case of execution time, if it does not gets a match from the resource pool it increases execution time, therefore cache based task scheduling was introduced to keep track of most used task in resource pool for subsequent processing and the network latency was improved based on best fit fog devices.

This result is also related with the work observed from[10], and in contrast with results obtained by[1] due to the introduction of deterministic spanning tree algorithm in the prolem formulation.

IV. CONCLUSION

Hybrid Meta-heuristics task scheduling algorithm for efficient latency in fog computing is proposed, the research combined meta-heuristic algorithm and deterministic spanning tree algorithm to achieve task scheduling in fog computing with shorter execution time, reduced latency and reduced energy consumption. The study proposed MPSO and MCSO for task allocation and virtual machine allocation between fog nodes and the user task or request, and hybridization of MPSO and SPT for task scheduling and resource allocation and management in collaboration between fog and cloud computing. The hybridization was to reduce the drawbacks of the heuristic or deterministic algorithms when used separately.

The research implemented the algorithms in iFogsim in which the performance of the algorithm was assessed, evaluated and then compared with other state of the art resource management and task scheduling algorithms in fog computing environment, our proposed method presented better results in term of resource execution time, load balancing, response time, execution cost and communication latency. From foregoing the research recommended implementation of the proposed method in other platforms that will incorporate mobility among fog devices like Mobfogsim simulator.

REFERENCES

[1]. M. Aliyu, M. M, A. Y. Gital, B. Souley, R. Kabir, M. A. Musa, F. U. Zambuk, J. C. Shawulu and I. M. Umar, "A Multi-Tier Architecture for the Management of Supply Chain of Cloud Resources in a Vitualized Cloud Environment: A Novel SCM Techniques for Cloud Resources Using Ant Colony Optimization and Spanning Tree," International Journal of Information Systems and Supply Chain Management (IJISSCM), p. 17, 2021.

[2]. S. Pradeep and Y. k. Sharma, "Effectual Secured Approach for Internet of Things with Fog Computing and Mobile Cloud Architecture Using Ifogsim," Proceedings of the World Congress on Engineering 2019, pp. 101-104, 2019.

[3]. I. M. M. Nasser, M. O. Al-Shawwa and S. Samy , "Artificial Neural Network for Diagnose Autism Spectrum disorder," International Journal of Academic Information Systems Research (IJAISR), pp. 27-32, 2019.

[4]. U. Sidiq, S. M. Aaqib and R. Ahm, "Diagnosis of Various Thyroid Ailments using Data Mining Classification Techniques," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 131-136, 2019.

[5]. A. N. Arbain and P. B. Y. Balakrishnan, "A Comparison of Data Mining Algorithms for Liver Disease Prediction on Imbalanced Data," International Journal of Data Science and Advanced Analytics, pp. 1-11, 2019.

[6]. A. K. Pal, P. Rawal, R. Ruwala and V. Patel, "Generic Disease Prediction using Symptoms with Supervised Machine Learning," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 1082-1086, 2019.

[7]. D. Sindhuja and R. J. Priyadarsini, "A Survey on Classification Techniques in Data Mining for Analyzing Liver Disease Disorder," International Journal of Computer Science and Mobile Computing, pp. 483-488, 2016.

[8]. F. Özyurt, T. Tuncer, E. Avci, M. Koç and I. S. glu, "A Novel Liver Image Classification Method Using Perceptual Hash-Based Convolutional Neural Network," Arabian Journal for Science and Engineering, pp. 3173-3182, 2019.

[9]. I. . M. Nasser and S. S. Abu-Naser, "Lung Cancer Detection Using Artificial Neural Network," International Journal of Engineering and Information Systems (IJEAIS), pp. 17-23, 2019.

[10]. T. R. Baitharua and S. K. Panib, "Analysis of Data Mining Techniques For Healthcare Decision Support System Using Liver Disorder Dataset," Procedia Computer Science (Science Direct), pp. 862-870, 2016.

[11]. H. Alahmera and A. Ahmeda, "Computer-Aided Classification of Liver Lesions from CT Images Based on Multiple ROI," Procedia Computer Science (Science Direct), pp. 80-86, 2016.

[12]. A. Akusok, D. Veganzones, Y. Miche, K.-M. Björk, P. d. Jardin , E. Severin and A. Lendasse, "MD-ELM: Originally Mislabeled Samples Detection using OP-ELM Model," Neurocomputing (Science Direct), pp. 242-250, 2015.

[13]. A. Abubakar, T. Mantoro, S. Moedjio, M. Anugerah Ayu, H. Chiroma, A. Waqas, . S. Muhammad Abdulhamid, M. Fatihu Hamza and A. Ya'u Gital, "A Support Vector Machine Classification of Computational Capabilities of 3D Map on Mobile Device for Navigation Aid," International Journal of Interactive Mobile Technologies (iJIM), pp. 4-10, 2016.

[14]. A. F. Ali and M. A. Tawhid, "A hybrid cuckoo search algorithm with Nelder Mead method for solving global optimization problems," Springer Plus, pp. 1-22, 2016.

[15]. T. Le Hoang , T. Son Hai and N. Thanh Thuy, "Image Classification using Support Vector Machine and Artificial Neural Network," I.J. Information Technology and Computer Science, pp. 32-38, 2012.

[16]. B. Bahrambeygi and H. Moeinzadeh, "Comparison of support vector machine and neutral network classification method in hyperspectral mapping of ophiolite mélanges–A case study of east of Iran," The Egyptian Journal of Remote Sensing and Space Sciences, pp. 1-10, 2017.

[17]. . J. Najmeh Sadat, S. Abdullah and M. Abdul Malek , "Master-Leader-Slave Cuckoo Search with Parameter Control for ANN Optimization and Its Real-World Application to Water Quality Prediction," Journal Pone, pp. 1-19, 2017.

[18]. M. Ehteram, V. P. Singh, A. Ferdowsi, M. Sayed Farhad , S. Farzin, H. Karami, N. Syuhadaa Mohd, H. Abdulmohsin Afan, H. L. Sai, . O. Kisi, M. A. Malek, . A. Najah Ahmed and A. El-Shafie, "An improved model based on the support vector machine and cuckoo algorithm for simulating reference evapotranspiration," Journal Pone, pp. 1-25, 2019.

[19]. G. Wang, "A Comparative Study of Cuckoo Algorithm and Ant Colony Algorithm in Optimal Path Problems," MATEC Web of Conferences, pp. 1-6, 2018.

[20]. X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications. Neural Computing and Applications," Neural Computing and Applications, pp. 1-11, 2014.

[21]. H. Chiroma, S. Abdul-kareem, A. Khan, N. M. Nawi, A. Y. Gital, L. Shuib, A. I. Abubakar, M. Z. Rahman and T. Herawan, "Global Warming: Predicting OPEC Carbon Dioxide Emissions from Petroleum Consumption Using Neural Network and Hybrid Cuckoo Search Algorithm," Journal Pone, pp. 1-21, 2015.

[22]. Z. He, K. Xia , W. Niu, N. Aslam and J. Hou, "Semisupervised SVM Based on Cuckoo Search Algorithm and Its Application," Hindawi Mathematical Problems in Engineering, pp. 1-14, 2018.

[23]. H. Fu and X. Liu, "PSO-Based Support Vector Machine with Cuckoo Search Technique for Clinical Disease Diagnoses," Hindawi Publishing Corporation the Scientific World Journal, pp. 1-8, 2014.

[24]. K. Duan, S. Fong, S. W. I. Siu, W. Song and S. S.-U. Guan, "Adaptive Incremental Genetic Algorithm for Task Scheduling in Cloud Environments," Symmetry, pp. 1-13, 2018.
[25]. G. Singh and M. Mahajan, "A Green Computing Supportive Allocation Scheme Utilizing Genetic Algorithm and Support Vector Machine," International Journal of Innovative Technology and Exploring Engineering (IJITEE), pp. 760-766, 2019.
[26]. H. Chuyan, "Service Cost of Resource Scheduling in Cloud Computing based on an Improved Algorithm Combining Support Vector Machine with Genetic Algorithm," International Journal of Grid and Distributed Computing, pp. 51-62, 2016.
[27]. K. Pradeep and T. P. Jacob, "CGSA scheduler: A multi-objective-based hybrid approach for task scheduling in cloud environment," Information Security Journal: A Global Perspective, pp. 77-91, 2018.
[28]. A. Kumari and M. Lipton, "A Review on Efficient Approach for Scheduling Tasks in Cloud Computing Environments," International Journal of Scientific Research & Engineering Trends, pp. 1550-1556, 2019.
[29]. I. M. Adamu , A. Y. Gital and S. Boukari, "A Hierarchical Framework for the Management of Virtualized Resources for Multi-Tier Services and Applications in Cloud Computing," International Journal of Advances in Scientific Research and Engineering (ijasre), Vol 4 (7), pp. 1-7, 2018.
[30]. S. Ghasemi and A. Hanani, "A Cuckoo-based Workflow Scheduling Algorithm to Reduce Cost and Increase Load Balance in the Cloud Environment," INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION, pp. 79-85, 2019.
[31]. S. WANG, Z. TIANYU and S. PANG, "Task Scheduling Algorithm Based on Improved Firework Algorithm in Fog Computing," IEEE Access, pp. 32385-32394, 2020.
[32]. X.-Q. Pham, N. . D. Man, N. D. T. Tri, N. Q. Thai and E.-N. Huh, "A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing," International Journal of Distributed Sensor Networks, pp. 1-16, 2017.
[33]. H. RAFIQUE, M. . A. SHAH, S. U. ISLAM , S. KHAN and C. MAPLE, "A Novel Bio-Inspired Hybrid Algorithm (NBIHA) for Efficient Resource Management in Fog Computing," IEEE Access, pp. 115760-115773, 2019.
[34]. M. Verma, N. Bhardwaj and A. K. Yadav, "Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment," International Journal for Information Technology and Computer Science, pp. 1-10, 2016.
[35]. T. Choudhari, "PRIORITIZED TASK SCHEDULING IN FOG COMPUTING," San Jose State University, San Jose, 2018.
[36]. G. Li, Y. Liu, J. Wu and D. Li, "Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing," MDPI Journal, pp. 1-16, 2019.
[37]. K. A. Kumari, J. K. R. Sastry and K. R. Rao, "Energy Efficient Load Balanced Optimal Resource Allocation Scheme for Cloud Environment," International Journal of Recent Technology and Engineering (IJRTE), pp. 146-153, 2019.
[38]. V. Ahari, . R. Venkatesan and D. P. p. latha, "A Survey on Task Scheduling using Intelligent Water Drops Algorithm in Cloud Computing," Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore, pp. 39-45, 2019.
[39]. S. Nazir, S. Shafiq, Z. Iqbal, M. Zeeshan, S. Tariq and N. Javaid, "Cuckoo Optimization Algorithm Based Job Scheduling Using Cloud and Fog Computing in Smart Grid," Springer Nature Switzerland, pp. 34-46, 2019.
[40]. A. M. Ahmed, T. A. Rashid and S. A. M. Soran, "Cat Swarm Optimization Algorithm: A Survey and Performance Evaluation," Hindawi Computational Intelligence and Neuroscience, p. 20, 2020.
[41]. K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder and B. Koldehofe, "Mobile Fog: A Programming Model for Large–Scale Applications on the Internet of Things," ACM, pp. 15-20, 2013.
[42]. C. Puliafito, D. M. Gonçalves, M. M. Lopes, L. L. Martins, E. Madeirab, E. Mingozzia, O. Ranac and L. F. Bittencourt, "MobFogSim: Simulation of

mobility and migration for fog computing," ELSEVIER Simulation Modelling Practice and Theory, pp. 1-25, 2020.

[43]. K. Matrouk and K. Alatoun, "Scheduling Algorithms in Fog Computing: A Survey," International Journal of Networked and Distributed Computing, pp. 59-74, 2021.

[44]. A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong and J. P. Jue, "All One Needs to Know about Fog Computing and Related Edge Computing Paradigms A Complete Survey," arXiv:1808.05283v3, pp. 1-48, 2019.

[45]. X.-S. Yang, "Analysis of Algorithms," in Nature-Inspired Optimization Algorithm, 2014, pp. 23-44.

[46]. D. Tian and Z. Shi, "MPSO: Modified particle swarm optimization and its applications," ELSEVIER SWEVO 352, pp. 1-46, 2018.

[47]. M. Aliyu, M. Murali, A. Y. Gital and S. Boukari, "Efficient Metaheuristic Population Based and Deterministic Algorithm for Resource Provisioning Using Ant Colony Optimization and Spanning Tree," International Journal of Cloud Applications and Computing, pp. 1-21, 2020.

[48]. D. M. Amancio da Silva, G. Asamooning, H. Orrillo, R. C. Sofia and P. M. Mendes, "An Analysis of Fog Computing Data Placement Algorithms," arXiv, pp. 1-8, 2020.

[49]. S. Kumar and D. S. Rana, "Various Dynamic Load Balancing Algorithms in Cloud Environment: A Survey," International Journal of Computer Applications (0975-8887), pp. 14-19, 2015.

[50]. R. Mahmud, K. Ramamohanaro and R. Buyya, "Latency-Aware Application Module Management for Fog Computing Environments," ACM Trans. Internet Technol., pp. 9-21, 2018.

[51]. U. Vadde and V. S. Kompalli, "Energy efficient service placement in fog Computing," peerj Computer Science, pp. 1-16, 2022.

[52]. S. M. Hussain and G. R. Begh, "Hybrid heuristic algorithm for cost-efficient QoS aware task scheduling in fog–cloud environment," Journal of Computational Science ELSEVIER, pp. 20-35, 2022.