OPTIMIZING RESOURCE UTILIZATION AND RESPONSE TIME IN CLOUD- FOG COMPUTING THROUGH TASK SCHEDULING ALGORITHM

Hardik Mahendrabhai Patel[1*], Dr. Kirit J. Modi[2]

Abstract—
An Analyzing the performance of a task scheduling algorithm in Cloud-Fog Computing is an important aspect of understanding the capabilities of the technology. By understand- ing the effectiveness of the scheduling algorithm, developers can make informed decisions about how to optimize their Cloud- fog computing applications. Task scheduling algorithms in Cloud-fog computing are used to allocate resources to tasks in order to achieve optimal performance measures such as Execution time throughput, and Number of nodes used. Different schedulingalgorithms can be used to address different performance re- quirements, and each algorithm has its own set of advantagesand disadvantages. Performance analysis can help identify which scheduling algorithm is best suited for a particular application, taking into account factors such as latency, node used, and task priorities. Additionally, performance analysis can provide insightinto the scalability of a task scheduling algorithm and how itcan be optimized for different workloads. Here we implement an algorithm which helps to execute a task in Cloud-fog environmentwith some basic parameter like time required, Number of flog node used Etc.

Index Terms—Cloud Computing, Fog Computing, Energy Efficient, Task Scheduling, Resource Utilization

[1*]PhD Scholar Computer Engineering Department, Sankalchand Patel Engineering Collage, Visnagar, Gujarat profhmp9@gmail.com
[2]Head Computer Engineering / Information Technology Department, Sankalchand Patel Engineering Collage, Visnagar, Gujarat, Kiritmodi@gmail.com

***Corresponding Author:** Hardik Mahendrabhai Patel
*PhD Scholar Computer Engineering Department, Sankalchand Patel Engineering Collage, Visnagar, Gujarat profhmp9@gmail.com

DOI: 10.48047/ecb/2023.12.si10.0064

I.s

II. INTRODUCTION

The Cloud-fog computing is a type of computing architec- ture that allows users to utilize remote computing resources that are accessed, such as storage and processors through a system and provided through service contributor. This means that users no need to acquire any equipment, as they can rent computational resources on an as-needed basis. The scalability and flexibility of cloud-fog computing are provided by theease of obtaining as organization's strategic increase, the cloud provider will provide more computational resources.

[1] Cloud-fog computing has a significant advantage in thatit minimizes the requirement for on-site hardware, resultingin reduced costs and less need for hardware maintenance.Furthermore, configuring and provisioning of local hardware is not necessary, freeing up time and resources for other tasks. As cloud-fog computing becomes more prevalent, the needfor additional hardware resources by cloud providers also increases, leading to a rise in energy consumption and asso- ciated costs. It is therefore imperative to minimize the energy consumption of these resources. In addition, high energy usage can impact the reliability of services by causing hotspots that affect performance due to increased operating temperatures. One approach to addressing this issue is the use of traditional energy-saving techniques, efficient airflow configurations and controlled fan systems within cooling systems are employed, and intelligent workload scheduling is implemented, whereby jobs are assigned or moved to servers to reduce energy consumption. This process, known as migration, is critical in current cloud-fog task scheduling as it helps to conserve time, energy and financial resources. Cloud computing, despite its many benefits, can suffer from problems related to latency, bandwidth, and data security. To address these issues, varia- tions of Several computing paradigms, such as fog computing and edge computing, have been developed as a result of cloud computing. [2] Fog computing is a network topology that aimsto bring cloud computing closer to end-users and devices by leveraging resources at the network edge. It can help address the issue of high latency by reducing the distance that data needs to travel between the user and the cloud. Additionally, fog computing can improve data security by keeping sensitive data closer to the source, reducing the risk of data breaches. The three-layer architecture is a framework designed for optimizing the scheduling of resources for IoT applications.It consists of the terminal layer, edge layer, and core layer,and it enables efficient allocation of resources and reduces communication and latency costs. The first level of scheduling involves the allocation of resources among fog groups, while the second level involves scheduling among fog nodes within the same fog group. A random scheduling approach is usedto manage IoT resources, but this does not consider the costof resource distribution and scheduling across fog clusters.To address this, a priority-based scheduling procedure has been implemented which uses the Optimize Response Time and Reconfigure Dynamic algorithms. This procedure can be used to maximize the utilization of available resources while minimizing communication and latency costs. [3] Fog-based network architecture with a multi-tiered structure has been proposed as a solution for the Internet of Everything (IoE) to reduce latency, resource usage, and energy consumption [4]. This approach utilizes a distributed computing model where computation and storage are moved closer to the edge devices, rather than relying solely on centralized cloud computing. Thisreduces latency and improves resource usage by allowing for more efficient processing and storage of data. [5]

Figure 1: Advantages of Fog Computing

It is crucial to mention that this strategy overlooks certain aspects, such as maintaining an even workload, organizing tasks, and ensuring data protection and confidentiality. When dealing with healthcare data, the quantity and speed at which IoT devices generate data can be overwhelming, and it isnecessary to devise measures to safeguard the isolation and defence of confidential patient information [6]. In order to reduce the costs and delays associated with communication, an approach that utilizes priority-based scheduling along with algorithms such as Optimize Response Time and Reconfig- ure Dynamic is

implemented. This method effectively distributes resources, optimizes their utilization, and minimizes the expenses related to communication and latency. [7]. An optimization problem that utilizes permutations can be used to formulate herculean tasks in definite time in a cloud-fog system. This approach takes into account the sequence in which real-time tasks are completed, which can enhance the performance of the scheduling algorithm. By using permuta- tions to represent the task order, the algorithm can find the optimal sequence that minimizes the overall completion time or maximizes the overall resource utilization [8]. It's important to note that this approach is not the only way to schedule tasksin a cloud-fog system, and other methods as priority-based scheduling and load balancing can also be used. However, itis an appropriate strategy when the system has many real-time tasks with tight deadlines. An auspicious technology Fog computing is intended for applications that demand extremely low latency. It bridges the gap between cloud computing and end-users by extending cloud computing functionality to the edge of the network through the use of fog nodes, which have computational, storage, and networking capabilities. Tasks thatare sensitive to latency can be processed by fog computingat the edge level, while tasks that can tolerate delay can be sent to the cloud. Traffic in the backhaul and core network is reduced, and energy consumption of the cloud is decreased. [9]The primary goal of this paper is for the reduction of energy consumption in the fog network while upholding the quality of service (QoS) standards required for low-latency applications.

To accomplish this, the paper suggests a solution based on three principles: resource allocation, traffic forecasting, and energy conservation. The solution proposed in this paper recommends splitting fog nodes into three categories: active, standby, and idle. The allocation of these nodes is determined dynamically based on the forecasted workload and the latency demands of the jobs. Only required number of fog nodes is utilized during a particular timeframe, while the additional fog nodes are placed in a low-influence state or sleep mode to save energy. The proposal put forward in this paper also involves managing the sleep mode of the message interface and dealing out unit separately for each fog node. The time of the sleep period is determined based on the assigned role of the fog node, such as actively working, on standby, or idle, and the necessary delay.

III. BACKGROUND THEORY AND RELATED WORK

In this section, we evaluate the existing algorithm work define by authors in various papers and talk about backgroundknowledge. Sundas Iftikhar et al HunterPlus is a novel re- source scheduling approach that leverages convolutional neural networks (CNNs) to enhance the performance of the existing GGCN scheduler, also known as HUNTER. The goal of HunterPlus is to optimize resource allocation in distributed computing systems, which involves efficiently scheduling tasksacross multiple compute nodes to maximize job completion rates while minimizing energy consumption. [3] However, the performance of HunterPlus is highly dependent on the dataset used for training. It was found by the authors that datasets generated from higher workload baselines were more effectivein training stable schedulers, suggesting that the choice of training data is crucial for achieving optimal performance.Overall, HunterPlus presents a promising approach for improving resource scheduling in distributed computing systems,and its performance highlights the potential of using CNNs to enhance existing scheduling models. Benila S et al The useof middle fog computing A way to has become increasingly popular in recent years as it is provided by perform dataprocessing and analysis closer to the edge of the network, reducing the time and cost associated with transmit large amounts of data to the cloud.

However, when dealing with time- sensitive applications, such as those in healthcare, the delay that is caused by transmitting data to the cloud andback can be intolerable. To address this issue, a Fog Managed Data Model with Weighted Fog Priority Job Scheduling and virtual machine allocation scheme has been proposed. [4] Moreover, the virtual machine allocation scheme optimizes theuse of resources by allocating virtual machines based on the requirements of each task, such as memory and processing power.

This ensures that the available resources are utilized efficiently and that the tasks are completed as quickly as possible. Overall, the Fog Managed Data Model with Weighted Fog Priority Job Scheduling and virtual machine allocation scheme provides an effective way to minimize the time lost while transmitting data to the cloud and back, making it suitable for time-sensitive applications such as those in healthcare. The proposed solution by Ali Kumar et al aimsto reduce energy consumption in the fog network while still meeting Service Level Agreement

(SLA) requirements forquality of service (QoS), especially the delay requirement for low-latency applications. It is suggested by the authors of the paper that keeping all fog nodes in an FCI (Fog Computing Infrastructure) active at all times to provide a requested service is not necessary. Instead, a dynamic approach is proposed to determine the number of active nodes in an FCI based on predicted demand for services. This method helps to reduce energy consumption while maintaining the QoS requirements specified by the SLA. [6] Overall, the proposed solution is a promising approach for reducing energy utilization in the fog network while meeting SLA requirements for QoS. By using a dynamic approach to identify the number of active nodes and implementing support nodes, solution can effectively balance energy efficiency and QoS requirements in a changingworkload environment. Husam Suleiman proposed The aim of the proposed framework is to manage fog job allocation and execution in a cloud-fog computing environment, taking into account the cost implications of different scheduling decisions.The risks of delays and energy consumption are balancedwith cost performance based on the QoS penalty of fog jobsin the framework's design.

Energy efficiency is aimed to be improved while maintaining QoS requirements [7] Overall,the proposed framework is a valuable contribution to the field of cloud-fog computing. By considering the expendi- ture implications of scheduling decisions and balancing them with QoS requirements, it provides a practical and effective approach to managing fog job allocation and execution in a cost-effective manner. This can help clients to achieve better cost performance while ensuring that QoS requirements are met. Mohamed Abdel-Basset et al A new hybrid evaluation algorithm for task scheduling has been proposed by the authors that combines two techniques to improve the efficiency ofthe scheduling process. The first technique involves using a scaling factor to calculate numerical values, while the second technique aims to achieve better results with less iteration. they have compared the proposed algorithm to existing algorithms. such as FCFS, SCA, and GWO after implementation, and the results indicate that In terms of scheduling efficiency, the new algorithm outperforms the existing ones [8]. Overall, proposedA promising contribution to the field of task scheduling in fog computing is made by the hybrid evaluation algorithm. By combining two techniques to improve scheduling efficiency and achieving better results with less iteration, the algorithm has the potential to enhance the performance of scheduling algorithms in fog computing applications. Sindhu V et al proposed The proposed scheduling algorithm, ECBTSA-IRA, is designed to balance energy efficiency, cost, and task per- formance effectiveness in fog computing environments. The algorithm represents the workload's task dependency as a directed acyclic graph (DAG) and prioritises and selects the workload that will be handled by each node, either a FogNode (FN) or the cloud, depending on the optimal efficiency factor.To select the best node for processing the workload, the algorithm balances schedule length, cost, and energy, taking into account energy consumption and cost, making it energy- cost-aware. An intelligent resource allocation technique is employed to balance the energy efficiency and cost of task execution in the FN or cloud, ensuring that the task is assigned to the resource with the lowest energy consumption and cost [10] Overall, the ECBTSA-IRA algorithm offers a comprehensive approach to fog job scheduling, taking into account multiple factors and trade-offs to optimize the use of resources while meeting QoS requirements. Kholoud Alatoun et al this approach to task scheduling in the IoMT is importantsince it can help to ensure that critical tasks are processed in a timely and energy-efficient manner. The framework takes into account multiple factor such as energy utilization, latency uti- lization to ensure optimal scheduling decisions. Additionally, the framework prioritises critical tasks that must be completed within a specific time frame, which is especially important in healthcare settings where timely and accurate processing of vital signs can be critical. However, as noted by the author, there are still limitations in the framework, and future work will need to focus on improving the framework's performance and expanding its capabilities to handle more types of vital signs and real-time applications beyond healthcare. Overall, this framework represents a promising step towards optimisingtask scheduling in the IoMT. [11] Zhong Zong proposed combination of two algorithm ant colony optimization and genetic algorithm and minimized the energy efficient of cloud computing data center. In this paper they define limitation can't consider resource allocation time and competition time.

[13] Sachi Gupta Proposed Multiprocessor task schedulingrefers to the execution of multiple tasks simultaneously in a system. Fog-cloud multiprocessor computing structures have gained significant popularity since their inception. However, similar to other networking systems, the

existing fog cloud system that relies on multiprocessor systems is not without its challenges. One of the major obstacles is scheduling, as there may be numerous clients and services that require resources, leading to energy consumption issues. A study was conducted to propose a hybrid approach that integrates GA and Ecs models to determine the optimal solution forefficient multiprocessor task scheduling. While most previous approaches focused on addressing energy-related issues, this study introduced the Hgecs method as part of the proposed approach to improve the overall scheduling efficiency. [17]

Mekala Ratna Raju Proposed Despite their advantages, fog computing devices have limited computation resources and power supply compared to cloud devices. This makes it challenging to execute tasks with strict deadlines and reduce the service latency and energy usage of fog resources, particularly in delay-sensitive applications of fog-enabled IoT architecture. To address this issue, the paper proposes an effective task scheduling strategy that allocates fog computing resources to IoT requests based on their deadline and resource availability, ensuring that the requests are completed within the deadline while minimizing energy consumption. Author proposes a task scheduling approach based on fuzzy reinforcement learning to minimize the average service time of tasks and reduce the energy consumption of fog nodes.

To achieve this goal, the authors formulate a mixed-integer nonlinear program for the task scheduling problem. [18] Shinu M. Rajagopal Proposed In Edge/Fog scenarios where critical medical IoT applications are deployed, patient privacy becomes a critical concern due to the dynamic and heterogeneous nature of the environment. To address these concerns, Federated Learning, a model trained on diverse data sources, can be used. In this paper,the authors propose the integration of Federated Learning into the distributed Edge-Fog- Cloud architecture of the IoT smart healthcare sector. They introduce FedSDM, a Federated Learning-based Smart Decision Making framework that uses ECG data in microservice-based IoT medical applications. [19] Elías Del-Pozo-Puñal Propsoed In order to model and analyze the behavior of Edge, Fog, and Cloud computing environments (such as power consumption, CPU usage, and bandwidth), itis necessary to use a simulation platform. However, scalabilityis an essential aspect of simulators, as they need to be able to add new components and simulate large infrastructures without compromising performance.

Many existing simulators cannot scale effectively as new elements are added to the system. Additionally, simulating mobile devices and knowing theirlocation is useful for developing and analyzing new algorithms in these environments. To address these issues, the ENIGMA simulator is presented and described in this article. ENIGMA is a scalable simulator of Edge, Fog, and Cloud computing infrastructures that can efficiently simulate a large numberof devices and elements and analyze various characteristics such as CPU usage, power consumption, network bandwidth, and application execution time. [20] The author proposes that Fog computing has become a prevalent solution for providing real-time computing services to terminal devices in intelligent production lines, thus solving the problem of transmission between cloud and terminal. However, when multiple services are offered by a single fog node, it becomes a challenge to select an optimal data processing path that minimizes latency and power consumption while ensuring the reliable transmission of data with varying priorities. According to the experimental results, the fog node task adaptive scheduling optimization algorithm not only reduced time latency but also saved power consumption. [21] The author suggests that the ease of service deployment and data management, as well as the provision of virtual resources, make it convenient to send computational or storage tasks to the cloud. In complex scenar-ios like smart city applications, where there are many tasks and virtual resources, task scheduling becomes a difficult problem to solve. To tackle this challenge, it is more efficient to use an automated task scheduling technique.

With the emergence of Fog Computing (FC) and Blockchain (BC) technologies, there are new opportunities to explore for automated task scheduling solutions. [22] In this Paper Author Proposed Fog node allocation strategy is designed specifically for power grids, andtakes into account the spatial distribution of data traffic sourceswithin the grid. It uses an unsupervised machine learning approach to determine the initial number and locations of Fog nodes, and then applies a reinforcement-based mechanism to minimize the required number of Fog nodes while stillmeeting the latency requirements of fixed scheduling and event-driven data services within the grid. The goal is to reduce capital costs while maintaining efficient data transmission and meeting latency requirements. [23] In this paper author proposes a lightweight task-scheduling framework from the

perspective of a cloud service provider for applications using both cloud and edge platforms. The framework addresses the challenge of efficiently utilizing both edge and cloud resources when necessary, while minimizing management overhead. Theauthors aim to answer two research questions: (i) how todistribute tasks to the edge resource pools and multi-clouds, and (ii) how to effectively manage these resource pools with low overheads. The framework is designed to quickly make decisions about task distribution and resource management.

[24] Malvinder Singh Bali Proposed It seems like the paper you are referring to propose a task prioritization and offloadingscheme for industrial-based sensor networks. The incoming tasks are assigned priorities and enqueued based on their priorities, and then offloaded to edge and cloud servers usinga multilevel feedback queue model. The proposed algorithm was evaluated against benchmark algorithms, and simulation results showed its effectiveness in terms of average queue delay, computational time, and energy consumption. The paperalso mentions that in the future, a technique will be developed to offload scheduled tasks to suitable computing devices, and the proposed framework will be compared with previous works to improve Quality of Service (QoS) requirements. [25]

IV. CLOUD – FOG ENVIRONMENT

Organizing tasks in cloud-based computing refers to the pro-cess of allocating and scheduling tasks in a cloud- environment for fog computing to optimize resource utilization and meet service-level objectives. Cloud-fog computing is a hybrid com-puting paradigm where the tasks are divided between cloud and fog computing resources based on their requirements and priorities. The objective of organizing tasks in cloud- based computing is to balance the workload between cloud and fog resources and to ensure that the most appropriate resourcesare used for each task. There are several algorithms use for task development in cloud- fog computing, including inactive scheduling, dynamic scheduling, in static scheduling and allo-cating tasks to resources before they are executed [6]. Tasks arerun-time assigned to resources in dynamic scheduling basedon the availability of resources. Hybrid scheduling combines the advantages of both static and dynamic scheduling. The cloud-fog computing environment presents several challenges to task scheduling, such as resource heterogeneity, uncertainty in resource availability, and dynamic changes in resource demands. To overcome these challenges, variousheuristics and optimization techniques have been proposed; People use ge- netic algorithms, ant colony optimization, and particle swarm optimization as meta heuristic optimization.

A. Scheduling the tasks for the Cloud-Fog computing environ-ment

To maximize resource utilization and achieve service-level goals, task allocation for the cloud computing refers to the process of allocating resources and scheduling tasks in a cloud computing environment. The purpose of task scheduling is meet service-level goals like response time, throughput, and re- liability while maximizing resource utilization and minimizing resource waste. [8]. There are several algorithms used in task scheduling, including static scheduling, dynamic scheduling, and hybrid scheduling. In static during scheduling, resources are allocated to specific tasks. Before they are executed. Dy- namic scheduling involves assigning tasks to resources at run-time, rather than pre-determining task assignments in advance based on the availability of resources. Hybrid scheduling com-bines the advantages of both static and dynamic scheduling [9].The cloud computing environment poses several challenges to task scheduling, such as resource heterogeneity, uncertainty in resource availability, and dynamic changes in resource demands. To overcome these challenges, various heuristics and optimization techniques have been proposed, in addition, various cloud service providers offer task scheduling services that can be used to schedule tasks for the cloud computing environment. These services accessed through APIs and used to manage resources and schedule tasks in a cloud computing environment. Overall, task scheduling is a main aspect of cloud computing and plays a crucial role in optimizing resource utilization, reducing resource waste, and meeting service-level objectives in a cloud computing environment [13]. Scheduling in fog computing involves the allocation for scheduling of tasks in a decentralized computing paradigm that distributes computing resources across the network's edge, closer to the data source devices. The main goal of task schedulingin fog computing is to reduce latency, conserve bandwidth, and improve quality of service for end-users by optimizing resource utilization and meeting service-level objectives. In fog computing, task scheduling involves the use of various algorithms such as static scheduling, dynamic scheduling, and hybrid scheduling. Static scheduling assigns tasks to resources before execution, while dynamic scheduling assigns tasks to available resources at runtime based on resource availability. Hybrid scheduling is a combination of

both static and dy- namic scheduling techniques. [5] Hybrid scheduling combinesthe advantages of both static and dynamic scheduling. The fog computing environment presents several challenges to task scheduling, such as resource heterogeneity, uncertaintyin resource availability, and dynamic changes in resource demands. To overcome these challenges, various heuristicsand optimization techniques have been proposed, In addition, various fog computing platforms offer task scheduling servicesthat can be used to tasks scheduling in a fog computing environment. These services can be accessed through APIs and used to manage resources and schedule tasks in a fog computing environment. Task scheduling in fog computing plays a crucial role in optimizing resource utilization, re- ducing resource waste, and meeting service-level objectives.It helps to minimize latency, reduce bandwidth usage, and Improving the quality of service for end-users is essential in fog computing, making it a crucial enabling technology for numerous (IoT) applications. In fog computing, the computing resources are distributed across the network's edge, closer to the data source devices, enabling faster processing and reducedlatency. This, in turn, enhances the overall performance of IoTand edge computing applications, providing a improved user experience. With the ability to perform real-time dispensation and reduce the burden on the cloud, fog computing is a key enabler for various use cases, including autonomous vehicles, smart homes, industrial automation, and many more [8].

V. IMPLEMENTATION AND RESULTS

According to the author [16], iFogSim, CloudSimSDN, and EdgeCloudSim are all offshoots of the popular cloud simulatorCloudSim and carry all of its benefits and drawbacks, includ- ing any known bugs. As a result, I favour using Cloudsim, a Java-based simulator.

Figure 2: User Request GUI

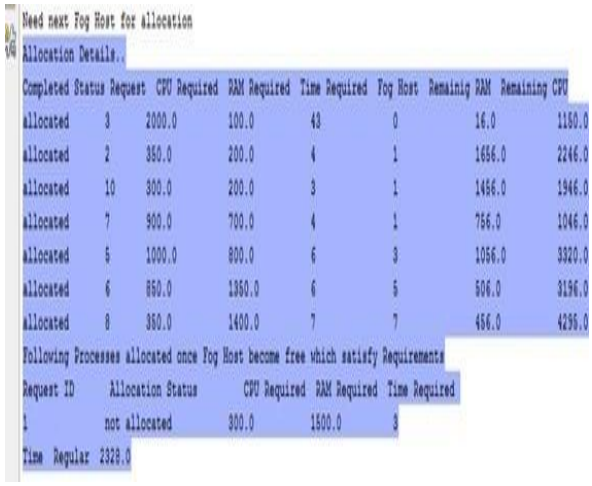

Figure 3: Number of User Process

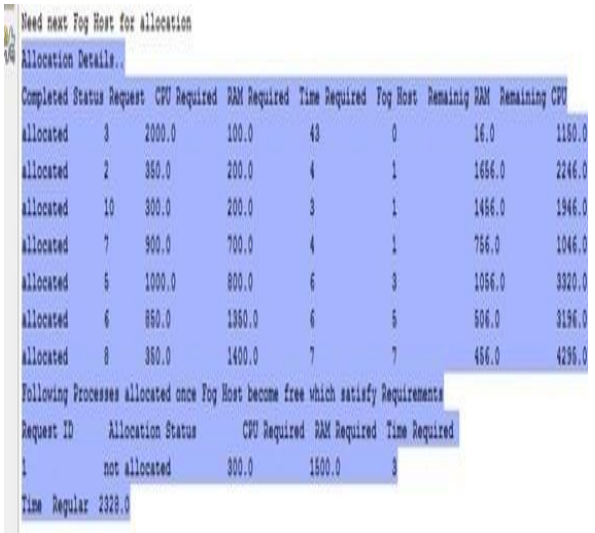

Figure 4: Fog Allocation Status

VMs - 50 Hosts - 30	No. of Hosts Used
No. of User Requests	Existing Method
5	2
10	4
15	4
20	5
25	6
30	7

Table I: Number of hosts with user requests

In Figure number 2 define the GUI of user request so user can enter the field of processing power and sent the request, figure number 3 consist total number of process are allocated, Figure number 4 consist once we allocated the process for execution we can identify that now how many RAM will be free so according we can assign the next process, and basedon that we will identify how many host are required as perthe user process request its define Table -1 and semantically for Table-2 how much time will be taken to executed the process. Here with, I have also shown the results for existing fog computing algorithms. Based on these

results, we will identify our proposed algorithm. In next section, I will show the proposed algorithm also.

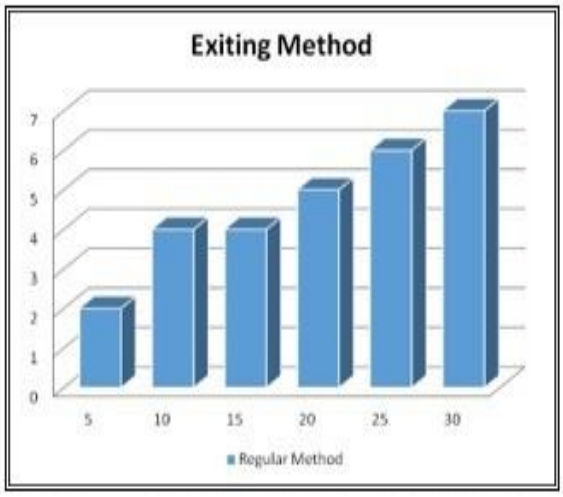

Figure 5: Compression between user request and hostused

VMs - 50 Hosts - 30	Time (in ms)
No. of User Requests	Existing Method
5	663
10	938
15	1121
20	1540
25	1842
30	2120

Table II: Time taken with respect to number of requests

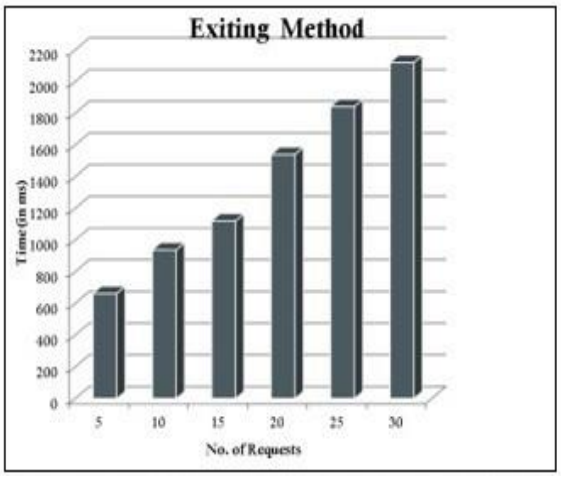

Figure 6: Compression between user request and hostused

VI. PROPOSED FLOW MODEL

This section describes in detail the algorithms required for the proposed model. Its define in two parts, the first algorithm define weather process are comfort for execution or not.

Second algorithm defines if priority process is coming then how we can handle. In the Figure number 5 consist once we allocated the process for execution will check the comfort first, and Figure number 6 consist will check the Priority if priority process are come will allocated the resource to them first for that purpose I have mention proposed algorithm steps.

Algorithm 1: - Check Comfort for Process allocationStep 1: Start
Step 2: Take a Task Request
Step 3: Assigns Priority (high or medium)
Step 4: send request to FMN
Step 5: Get the status of FMN
Step 6: Check for comfort
Step 7: if comfort is good then go to step 8 if Comfort is not Good go to End it
Step 8: Add into Probable list and sort list as per Comfort
Step 9: if is compatible to access the request go to
step 10.if not Compatible to access the request Go to step 11
Step 10: Allocate on fog node and Update status and Generate Final Allotment list and go to step 12.
Step 11: Check next in Probable list if yes then go to
step10. if no then Forward request to the cloud and go to step12.
Step 12: End

Algorithm 2: -Process Request Priority wise allocationalgorithm
Step 1: Start
Step 2: Receive a task request
Step 3: Assign priorities to the request (high, medium, low)
Step 4: Load the request into a queue based on its priority
Step 5: Select the top request from the high-priority queue
Step 6: Check if a fog node is available to process the request
Step 7: If no fog node is available, move to the next fognode
Step 8: If there are no more fog nodes, forward the requestto the cloud
Step 9: If the fog node is the last node, forward the requestto the cloud
Step 10: If the fog node is not the last node, return to step6
Step 11: Generate final statistics
Step 12: End

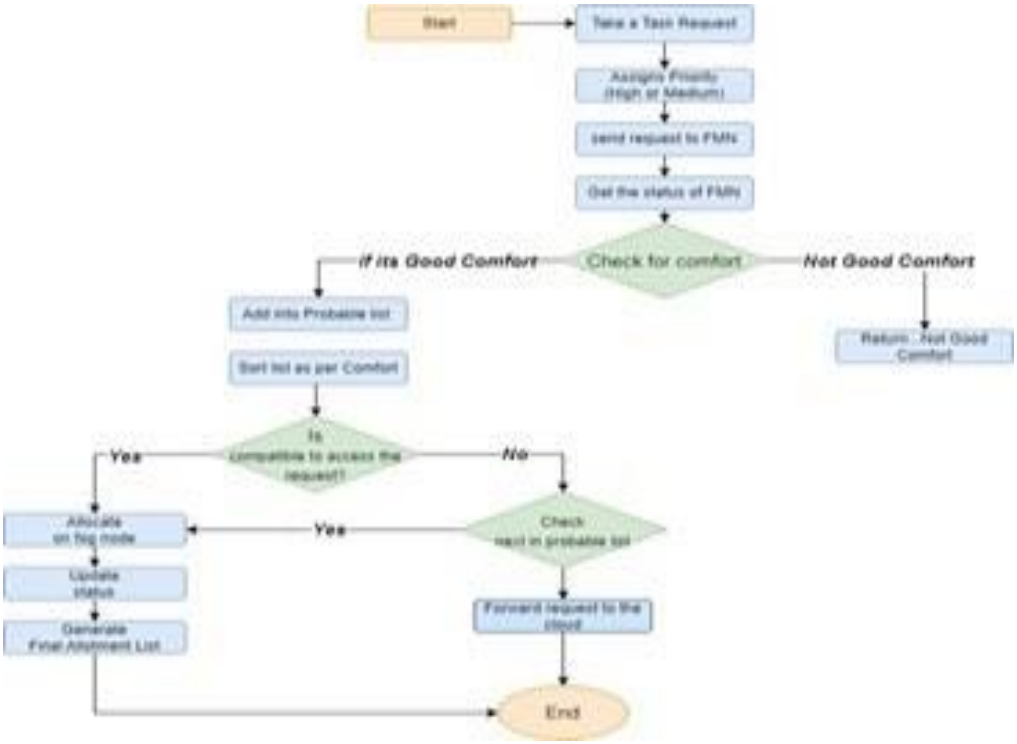

Figure 7: Check Comfort for Process allocation

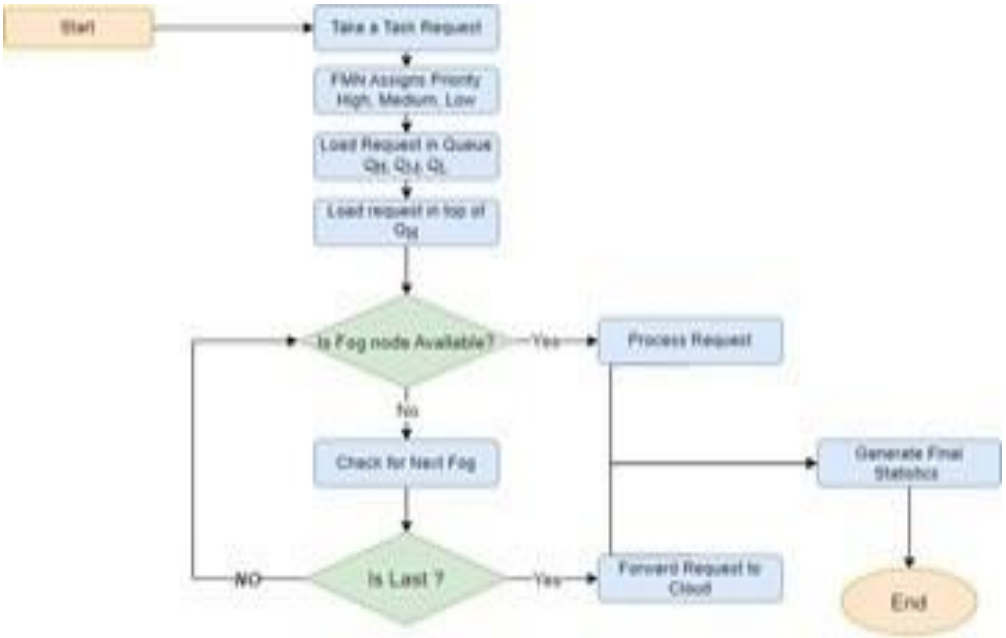

Figure 8: Assign Priority for Process

VII. CONCLUSION

Overall, task scheduling algorithms play a crucial role in enhancing the efficiency and quality of Cloud-fog computing services. They improve resource utilization, decrease service latency, and enhance overall system efficiency. Additionally, task scheduling algorithms enhance the security of fog com- puting systems by enabling task execution in a different environment. Furthermore, the development of task schedulingalgorithms is an ongoing process, with new algorithms being developed each year to improve upon existing ones. Overall, task scheduling algorithms are an important part of the Cloud- fog computing architecture, and their effective use can helpto improve the performance and reliability of fog computing services. With this number of important parameter that con- sider for checking of effectiveness of Cloud-fog environment is number of nodes used, time required for Execution Etc. For future work we try to propose a model which improvethe existing parameter and provide better efficient Cloud-fog model to the user.

REFERENCES

1. Pierangela Samarati, Latanya Sweeney" Generalizing Data to Provide Anonymity when Disclosing Information".
2. Nair, Biji, and S. Mary Saira Bhanu. "Task Scheduling in Fog Node Within the Tactical Cloud." Defence Science Journal, vol. 72, no. 1, Defence Scientific Information and documentation Centre, Jan. 2022, pp. 49–55. Crossref, https://doi.org/10.14429/dsj.72.17039.
3. Iftikhar, Sundas, et al. "HunterPlus: AI Based Energy-efficient Task Scheduling for Cloud–fog Computing Environments." Internet of Things, vol. 21, Elsevier BV, Apr. 2023, p. 100667. Crossref, https://doi.org/10.1016/j.iot.2022.100667.
4. Benila S, Benila S., and Usha Bhanu N. Benila S. "Fog Man-aged Data Model for IoT Based Healthcare Systems.", vol. 23, no. 2, Angle Publishing Co., Ltd., Mar. 2022, pp. 217–26. Crossref, https://doi.org/10.53106 /160792642022032302003.
5. Abohamama, A. S., et al. "Real-Time Task Scheduling Algorithm for IoT- Based Applications in the Cloud–Fog Environment." Journal of Network and Systems Management, vol. 30, no. 4, Springer Science and Business Media LLC, July 2022. Crossref, https://doi.org/10.1007/s10922-022- 09664-6.
6. Pg. Ali Kumar, Dk. Siti Nur Khadhijah, et al. "Green Demand Aware Fog Computing: A Prediction-Based Dynamic Resource Provisioning Approach." Electronics, vol. 11, no. 4, MDPI AG, Feb. 2022, p. 608. Crossref, https://doi.org/10.3390/electronics11040608.
7. Suleiman, Husam. "A Cost-Aware Framework for QoS-Based and Energy-Efficient Scheduling in Cloud–Fog Computing." Future In- ternet, vol. 14, no. 11, MDPI AG, Nov. 2022, p. 333. Crossref, https://doi.org/10.3390 /fi14110333.
8. Abdel-Basset, Mohamed, et al. "Task Scheduling Approach in Cloud Computing Environment Using Hybrid Differential Evolution." Math- ematics, vol. 10, no. 21, MDPI AG, Oct. 2022, p. 4049. Crossref, https://doi.org/10.3390/math10214049.
9. Perez Abreu, David, et al. "A Comparative Analysis of Simulators for the Cloud to Fog Continuum." Simulation Modelling Practice and Theory, vol. 101, Elsevier BV, May 2020, p. 102029. Crossref, https://doi.org/10.1016 /j.simpat.2019.102029.
10. V, Sindhu, et al. "Energy-Efficient Task Scheduling and Resource Allocation for Improving the Performance of a Cloud–Fog Environment." Symmetry, vol. 14, no. 11, MDPI AG, Nov. 2022, p. 2340. Crossref, https://doi.org/10.3390/sym14112340.
11. Alatoun, Kholoud, et al. "A Novel Low-Latency and Energy-Efficient Task Scheduling Framework for Internet of Medical Things in an Edge Fog Cloud System." Sensors, vol. 22, no. 14, MDPI AG, July 2022, p. 5327. Crossref, https://doi.org/10.3390/s22145327.
12. Kak, Sanna Mehraj, et al. "Task Scheduling Techniques for Energy Efficiency in the Cloud." EAI Endorsed Transactions on Energy Web, vol. 9, no. 39, European Alliance for Innovation n.o., June 2022, p. e6. Crossref, https://doi.org/10.4108/ew.v9i39.1509.
13. V, Sindhu, et al. "Energy-Efficient Task Scheduling and Resource Allocation for Improving the Performance of a Cloud–Fog Environment." Symmetry, vol. 14, no. 11, MDPI AG, Nov. 2022, p. 2340. Crossref, https://doi.org/10.3390/sym14112340.
14. Sing, Ranumayee, et al. "EMCS: An Energy-Efficient Makespan Cost- Aware Scheduling Algorithm Using Evolutionary Learning Approach for Cloud-Fog-Based IoT Applications." Sustainability, vol. 14, no. 22, MDPI AG, Nov. 2022, p. 15096. Crossref, https://doi.org/10.3390/su142215096.
15. Zhong Zong. An Improvement of Task Scheduling Algorithmsfor Green Cloud Computing" The 15th International Conference on Computer Science Education (ICCSE 2020)
16. David Perez Abreu et al "A Comparative Analysis of Simulators for the Cloud to Fog Continuum." - ScienceDirect, 23 Nov. 2019, https://doi.org/10.1016/j.simpat.2019.102029.
17. Sachi Gupta et al "Multiprocessor task scheduling using multi-objective hybrid genetic Algorithm in Fog-cloud computing "Science Direct, 13 April. 2023, https://doi.org /10.1016/j.knosys.2023.110563.
18. Mekala Ratna Raju et al "Delay and energy aware task scheduling mechanism for fog-enabled IoT applications: A reinforcement learning approach" Volume 224, April 2023, 109603, https://doi.org /10.1016/j.comnet.2023.109603.
19. Shinu M. Rajagopal et al ""FedSDM: Federated Learning Based Smart Decision Making Module for ECG Data in IoT Integra ted Edge–Fog–Cloud Computing Environments." - ScienceDirect, 14 Apr. 2023, https://doi.org/10.1016/j.iot.2023.100784.
20. Elías Del-Pozo-Puñal et al "A Scalable Simulator for Cloud, Fog and Edge Computing

Platforms with Mobility Support." – Science Direct, 6 Mar. 2023, https://doi.org/10.1016/j.future.2023.02.010.
21. Fulong Xua et al "Adaptive Scheduling Strategy of Fog Computing Tasks with Different Priority for Intelligent Production Lines – Science Direct, 19 Apr. 2021, https://doi.org/10.1016/j.procs.2021.02.064.
22. Hamza Baniata et al "PF-BTS: A Privacy-Aware Fog-enhanced Blockchain-assisted Task Scheduling - ScienceDirect, 30 Sept.2020, https://doi.org/10.1016/j.ipm.2020.102393.
23. Muhammad Ali Jamshed et al "Reinforcement Learning-based Alloca- tion of Fog Nodes for Cloud-based Smart Grid" - ScienceDirect, 18 Mar. 2023, https://doi.org/10.1016/j.prime. 2023. 100144.
24. Thomas Dreibholz et al" Towards a Lightweight Task Scheduling Framework for Cloud and Edge Platform" - ScienceDirect, 11 Dec. 2022, https://doi.org/10.1016/j.iot.2022. 100651.
25. Malvinder Singh Bali et al "An Effective Technique to Schedule Priority Aware Tasks to Offload Data on Edge and Cloud Servers" - ScienceDirect, 10 Jan. 2023, https://doi.org /10.1016/j.measen.2023.100670