Task offloading as Optimization Problem over IoT Network: A Review

Suraj Kumar[1], Rakesh Shivhare[2]
[1]Department of CSE, Radharaman Engineering College, Bhopal (M.P.), India
[2]Department of CSE, Radharaman Engineering College, Bhopal (M.P.), India
[1]suraj543kum@gmail.com

* Corresponding Author: Suraj Kumar

Abstract: *The development of complex services is being facilitated by advancements in networking technologies including 4G long-term evolution (LTE), wireless broadband (WiBro), low-power wide area networks (LPWAN), 5G, LiFi, and others. Along with increased computation, connectivity, and cognitive capabilities, there are more online apps available. Although the technology currently in use is becoming more powerful in terms of features and capabilities, it is still unable to carry out smart, autonomous, and intelligent tasks like those frequently needed for smart healthcare, ambient assisted living (AAL), virtual reality, augmented reality, intelligent vehicular communication, as well as many services related to smart cities, the Internet of Things (IoT), the tactile internet, and the Internet of Vehicles. We give a taxonomy of recently proposed offloading strategies for areas like fog, cloud computing, and IoT. Additionally, we go through the middleware technologies that allow offloading in cloud-IoT applications as well as the variables that are crucial for offloading in a certain situation. We also discuss potential areas for future research in edge and fog computing. Keywords: Internet of Things (IoT), middleware, offloading, cloud computing, fog computing, and edge computing.*

Keywords:.

I. Introduction

The Internet of Things (IoT) is the networking of physical items with electronics built into its architecture to enable communication and the detection of interactions between them or with the surrounding environment. IoT-based technology will provide higher levels of services in the future years, effectively altering how individuals go about their daily lives. Just a few categories where IoT is well established include improvements in medicine, power, gene therapies, agriculture, smart cities, and smart homes. Smart homes, healthcare, traffic management, smart transportation, video surveillance, and emergency response are just a few of the many IoT application fields. Because the cloud can provide enormous storage and processing, cloud computing is used as a backend layer for data storage and analysis.Task offloading is what this is. Fabrizio Marozzo served as the assistant editor who oversaw the assessment of this submission and gave his approval for publication. where the IoT devices' heavy workload, or the volume of data they produce, is sent to the cloud for processing[1]. Offloading jobs to the cloud layer is ineffective, though, as this will increase network bandwidth usage and most of the data can be censored due to high redundancy. Furthermore, because of the relatively high network latency, shifting IoT tasks to the cloud causes a delay in the data analysis's response time. In order to bring computing capabilities closer to IoT devices, the fog computing idea was created. [2]

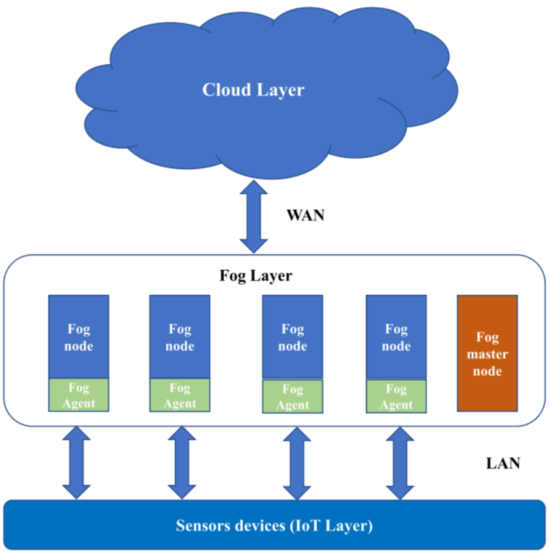

Figure 1. Layered fog computing infrastructure

IoT Task Offloading to Fog Computing

By utilising low network latency, IoT task offloading to fog computing as opposed to the cloud avoids excessive network congestion and speeds up data analysis reaction times. Because it offers computational capabilities, such as faster response times and reduced latencies, at the edge of the network close to the IoT devices, this idea is also known as edge computing. However, fog computing and edge computing are separate from one another. Fog computing uses network devices like routers and gateways inside the LAN of the IoT devices to deliver computing, networking, and storage services to those devices. In contrast, edge computing uses tiny data centres connected to WiFi access points near IoT devices to offer computation and storage services. [3]Specific smart applications, such autonomous driving, health care apps, and traffic control, require a certain quality of service (QoS), which fog computing offers. The right fog nodes must be carefully chosen for task offloading, and jobs must be evenly distributed among the nodes while still meeting the tasks' QoS requirements, which include their response times. By using low latency, shifting IoT tasks from the cloud to fog computing reduces the response time for data analytics.Due to the necessity of providing acceptable QoS due to the IoT linked devices' rapid growth rate, fog computing has intelligently adopted IoT device improvement while enhancing cloud computing's capabilities. [4] Fog Computing sells Internet of Things (IoT) devices that include LAN computing, networking, and storage capabilities and make use of network devices like routers and gateways. To improve computation quality and QoS, choosing the right fog node is crucial. Fog computing meets the QoS requirements for smart cities, smart energy, healthcare, cars, agriculture, and other applications. Cloud computing's drawbacks are resolved by fog computing.
In order to attain the necessary smartness, intelligence, and autonomy, many services (such online gaming and image processing) increasingly demand the integration of artificial intelligence (AI). We must effectively manage the execution of more complicated activities based on the requirements of the application as apps get more complex and smarter. The resource demands of some of these jobs could be much higher than what the end-device user's can handle. The duties are delegated in certain situations to the middleware, which may then delegate them to the cloud. It is important to remember that tasks are not restricted to computation alone, but also to other resources like storage.

The main contributions of this work are:

• We present the various criteria used by recently proposed middleware technologies when tasks are offloaded in the fog computing environment.

• We identify future research challenges that still need to be addressed to improve tasks' offloading performance, efficiency, and reliability in fog computing

II. LITERATURE REVIEW

In order to effectively load balance IoT tasks over the fog nodes while taking into account communication costs and response times, Hussein et al.[1] propose two different scheduling algorithms using two nature-inspired meta-heuristic schedulers, ant colony optimization (ACO) and particle swarm optimization (PSO). The round robin (RR) algorithm's experimental results are contrasted with those of the suggested algorithms. The evaluations demonstrate that the proposed ACO-based scheduler effectively load balances the fog nodes and improves IoT application response times when compared to the proposed PSO-based and RR algorithms.

The other two variations were proposed by Basset et al. [2]. Modified MPA (MMPA) is the name of the first iteration, which modifies MPA to improve their exploitation capabilities by using the most recent positions rather than the most recent best one. The second one will increase MMPA by using a ranking strategy based reinitialization and mutation toward the best, as well as reinitializing half of the population at random after a set number of iterations to get rid of local optima and mutating the remaining half in favour of the best-so-far solution. Based on different performance measures, including energy consumption, makespan, flow time, and carbon dioxide emission rate, the three versions are suggested along with additional metaheuristic and genetic algorithms. The enhanced MMPA may perform better than all other algorithms and other methods.

A scalable algorithm for offloading time-sensitive tasks via a semi-network aware distributed scheduling mechanism is presented by Ataie et al. in their paper [3]. The proposed method, on average, outperforms the state-of-the-art based on evaluation results for acceptance rate, response time, and network resource usage.

A school attendance system is presented by Sharaf et al. [4] that encourages the use and collaboration of the digital continuum while avoiding excessive bandwidth consumption and significant latency. The following contributions are made by this work: IoTs and fog are used in a unique way in (a) to improve the school absence tracking system. It advocates transparent workload offloading across the IoTs, edge, HPC, big data, Artificial Intelligence (AI), and cloud layers of the digital continuum using two well-known protocols, MQTT and TCP; and (b) it presents a case study that makes use of the digital continuum to complete an interesting task.

An overview of RL applications to address issues with resource allocation in the fog computing environment is provided by Dang et al. in their publication [5]. The unresolved problems and difficulties are examined and discussed for future research.

According to Gedawy et al[6] .'s MAESTRO system, users can delegate computing tasks to a number of nearby FemtoClouds. We provide an integrated architecture for MAESTRO that consists of two fresh scheduling methods for distributing computing workloads among femtoclouds. Each of our scheduling methods is made to help the system function more effectively whether the network infrastructure is strong or weak. To evaluate our system's performance on our experimental testbed, we put into practise a complete prototype. The results show that our specialised scheduler outperforms state-of-the-art schedulers by up to 55% in environments with poor communication, while our other specialised scheduler outperforms state-of-the-art schedulers by up to 67% in environments with good communication.

According to Maswood et al. [7], an integrated fog-cloud system can support real-time applications at a cheaper operational cost by minimising resource costs and cutting back on delays. In order to offer a unique optimization model in this context, we first show a collaborative three-layer fog-cloud computing environment. At first, the demand from the traffic generating sensors is still being met by layer-1 fog's regional capacity as minimising bandwidth costs takes precedence. The remaining need is met by layer-1 fog from other regions, layer-2 fog from other regions, and lastly cloud if the demand for a region exceeds its capacity. However, demand is divided among various resources to minimise delay when load balancing is the top concern.

Kishor et al.[8] To load the IoT-sensor applications jobs in a fog environment, Kishor et almeta-heuristic .'s scheduler Smart Ant Colony Optimization (SACO) task offloading technique is provided. Round Robin (RR), throttled scheduler algorithm, and two bio-inspired algorithms, such as modified particle swarm optimization (MPSO) and Bee life algorithm, are used to compare the performance of the suggested approach (BLA). The proposed Smart Ant Colony Optimization (SACO) algorithm significantly reduces latency when compared to Round Robin (RR), throttled, MPSO, and BLA in numerical results for task loading of IoT-sensor applications. In comparison to Round Robin (RR), throttled, MPSO, and BLA, the proposed technique shortens task loading time by 12.88, 6.98, 5.91, and 3.53%.

The multi-objective task scheduling issue of intelligent production lines is the subject of Z.Yin et al. [9], who also provide a task scheduling method based on work priority. First, we constructed a multi-objective function for task scheduling that minimises the service delay and energy consumption of the tasks and set up a cloud-fog computing architecture for intelligent manufacturing lines. The best work scheduling strategy is also looked for using the enhanced hybrid monarch butterfly optimization and enhanced ant colony optimization algorithm (HMA). Finally, thorough simulation studies are used to evaluate HMA, demonstrating that it outperforms other algorithms in terms of task completion rate. The completion rate of all tasks is greater than 90% when the number of nodes exceeds 10, which effectively satisfies the real-time requirements of therequirements of the corresponding tasks in the intelligent production lines. In addition, the algorithm outperforms other algorithms in terms of maximum completion rate and power consumption.

Multi-objective Pareto ant colony optimization is the approach Huang et al.[10] suggest using to optimise ant colonies (MRPACO). On MRPACO, we have carried out a lot of experiments. The experimental results demonstrate that our approach yields solutions that are qualified in terms of both diversity and accuracy, which are the primary metrics used to assess a multiobjective algorithm.

Table 1.Comparsion of techniques used in fog computing

Author name	Technique	Advantages	Limitations of the techniques
Hussain et al.[1]	Ant Colony Optimization	provides a significant improvement in IoT application response times and effectively balances the tasks over the fog nodes.	Used only single point connection between cloud and fog
Basset et al.[2]	MMPA	Minimized the cost offloading the task from IoT applications	Communication delay is not considered for reduction of cos
Dang et al.[5]	overview of RL applications	Reduced delay for IoT applications	Computation and communication delay is not discussed
Gedawy et al.[6]	MAESTRO	Optimized the energy for offloading the task in fog	Not discussed the communication delay in optimizing the energy

		environment and considered the computational delay	
Kishor et al.[8]	Ant Colony Optimization (SACO)	Energy consumption minimized	Only single user system model is used

III. CHALLENGES IN FOG IOT

We explore some of the issues that will need to be resolved in the future to make task offloading in fog computing seamless, effective, and dependable and to deliver high end-to-end performance and low latency (as well as quick reaction) for a variety of fog-based applications.

Allocating resources and scalability
Finding the appropriate number of resources needed at the site where the tasks will be carried out is one of the main issues associated with offloading.
It is common knowledge that if we have more resources than we need, some of them may go unused, and if we have fewer resources than we need, offloading may occur too frequently. Depending on the service, there will be a resource trade-off.

Scalability of application
It is expected that a specific number of users will utilise the application when it is deployed. The capacity of the background algorithms to process additional requests will determine the capacity to handle more users. As we previously discussed, a variety of parameters, including load balancing, energy efficiency, and other factors, are used in the process that performs the offloading. Therefore, it is still unclear how much the application should be scaled up to ensure that the offloading task is not impacted.

Service level agreement and compatibility of services
Making sure that the original service level agreement (SLA) is not broken while offloading remains a concern since offloading includes at least two parties: the tasks that need to be offloaded and the service provider where these duties are offloaded. In this situation, it is crucial to take into account SLA matching, SLA monitoring, and violation. Different devices would be running various algorithms or types of software, and they would likely carry out tasks in various ways that might not be compatible with one another (for example, meeting the latency requirement).

Security and privacy of data and users
The risk of data theft and misuse increases as more data hops are involved in unloading. Data communication and storage are increasingly vulnerable to hacking and theft when several nodes and systems interact with one another. As a result, effective and strong data security measures would be necessary. Reliable parties would also need to be involved, which would necessitate the definition and use of trustworthiness requirements.

IV. CONCLUSION

The workload prediction and best offloading choice for a cloud network for fog computing are proposed in this work. By giving cloud-like compute and storage capabilities at reduced reaction time and energy consumption, the result analysis will improve the processing capacities of network nodes and IoT goods.

REFERENCES

[1] M. K. Hussein and M. H. Mousa, “Efficient Task Offloading for IoT-Based Applications in Fog Computing Using Ant Colony Optimization,” IEEE Access, vol. 8, pp. 37191–37201, 2020, doi: 10.1109/ACCESS.2020.2975741.

[2] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, and N. Kumar, “Energy-Aware Marine Predators Algorithm for Task Scheduling in IoT-Based Fog Computing Applications,” IEEE Trans. Ind. Informatics, vol. 17, no. 7, pp. 5068–5076, 2021, doi: 10.1109/TII.2020.3001067.

[3] I. Ataie, T. Taami, S. Azizi, M. Mainuddin, and D. Schwartz, “D2FO: Distributed Dynamic Offloading Mechanism for Time-Sensitive Tasks in Fog-Cloud IoT-based Systems,” in 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC), 2022, pp. 360–366. doi: 10.1109/IPCCC55026.2022.9894304.

[4] M. Sharaf, “Smart Schools: Using IoTs and Fog Computing to Predict Underperformance,” in 2020 2nd International Conference on Computer and Information Sciences (ICCIS), 2020, pp. 1–6. doi: 10.1109/ICCIS49240.2020.9257689.

[5] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq, and D.-S. Kim, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," J. Commun. Networks, vol. 24, no. 1, pp. 83–98, 2022, doi: 10.23919/JCN.2021.000041.

[6] H. Gedawy, A. Elgazar, and K. A. Harras, "MAESTRO: Orchestrating Computational Offloading to Multiple FemtoClouds in Various Communication Environments," IEEE Access, vol. 10, pp. 27096–27112, 2022, doi: 10.1109/ACCESS.2022.3152075.

[7] M. M. Shahriar Maswood, M. D. R. Rahman, A. G. Alharbi, and D. Medhi, "A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in a Cooperative Three-Layer Fog-Cloud Computing Environment," IEEE Access, vol. 8, pp. 113737–113750, 2020, doi: 10.1109/ACCESS.2020.3003263.

[8] Kishor and C. Chakarbarty, "Task Offloading in Fog Computing for Using Smart Ant Colony Optimization," Wirel. Pers. Commun., pp. 1–22, 2021, doi: 10.1007/s11277-021-08714-7.

[9] Z. Yin et al., "A Multi-Objective Task Scheduling Strategy for Intelligent Production Line Based on Cloud-Fog Computing," Sensors, vol. 22, no. 4, 2022, doi: 10.3390/s22041555.

[10] T. Huang, W. Lin, C. Xiong, R. Pan, and J. Huang, "An Ant Colony Optimization-Based Multiobjective Service Replicas Placement Strategy for Fog Computing," IEEE Trans. Cybern., vol. 51, no. 11, pp. 5595–5608, 2021, doi: 10.1109/TCYB.2020.2989309.