An Adaptive and Efficient Scheduling Framework for Decentralized Fog Environments: Design and Problem Formulation

Gurpreet Singh Chhabra[1], Subhendu Kumar Pani[2]

[1]Postdoctoral Researcher, Lincoln University College, 47301, Petaling Jaya, Selangor Darul Ehsan, Malaysia;

[2]Department of Computer Science and Engineering, Krupajal Engineering College, Biju Patnaik University of Technology (BPUT), Odisha, India.

Email IDs: pdf.gurpreet@lincoln.edu.my, skpani.utkal@gmail.com.

Abstract: The increasing demand for low-latency and energy-efficient processing in Internet of Things (IoT) applications has highlighted the limitations of centralized scheduling in fog computing environments. This paper proposes an adaptive and efficient scheduling framework for decentralized fog computing, where local schedulers at each fog node make real-time decisions based on dynamic resource conditions. A cooperative scheduling mechanism among neighboring nodes is introduced to improve resource utilization and reduce cloud dependency, with the cloud used only as a fallback layer. The scheduling problem is formulated as a multi-objective optimization model considering latency, energy consumption, and load balancing. An adaptive cost-based approach with learning-driven weight adjustment enables the system to dynamically respond to changing workloads. The proposed framework provides a scalable, decentralized, and efficient solution for fog computing environments.

Keywords: Decentralized Fog Computing; Adaptive Scheduling Framework; Cooperative Scheduling; Multi-objective Optimization; Resource Management.

Introduction

The rapid growth of Internet of Things (IoT) applications has intensified the demand for low-latency and energy-efficient data processing, positioning fog computing as a key paradigm for distributed computation. By extending cloud capabilities closer to end devices, fog computing reduces communication delays and alleviates network congestion. However, efficient task scheduling in fog environments remains a complex challenge due to resource heterogeneity, dynamic workloads, and the lack of centralized control mechanisms.

To address these challenges, several approaches have been proposed in recent literature. Reinforcement learning (RL)-based scheduling techniques have demonstrated the ability to adapt to dynamic environments and improve system performance [2][4]. Additionally, multi-objective optimization methods have been explored to balance critical performance metrics such as latency, energy consumption, and resource utilization [3]. Decentralized scheduling strategies further enhance scalability and robustness by enabling local decision-making at fog nodes [5].

More recently, federated learning (FL) has emerged as a promising solution for enabling collaborative intelligence across distributed nodes while preserving data privacy [1][6]. Despite these advancements, most existing approaches either rely on isolated decision-making or lack efficient cooperation among fog nodes, limiting their effectiveness in highly dynamic environments. To overcome these limitations, this paper proposes an adaptive, decentralized scheduling framework that enables cooperative node interaction, aiming to improve scalability, resource utilization, and real-time decision-making in fog computing systems.

Proposed Methodology

- *System Architecture*:

  To address the limitations of centralized scheduling in fog environments, this work proposes a decentralized, adaptive scheduling framework deployed across a three-layer architecture comprising the Edge Layer, Fog Layer, and Cloud Layer.

  At the Edge Layer, heterogeneous IoT devices such as sensors and mobile systems continuously generate tasks with diverse computational and latency requirements. Each task is represented as ($T_i = (C_i, D_i, S_i, \delta_i)$), where ($C_i$) denotes computational demand, (D_i) represents data size, (S_i) indicates memory requirement, and (δ_i) is the deadline constraint. Due to limited resources at the edge, tasks are offloaded to nearby fog nodes.

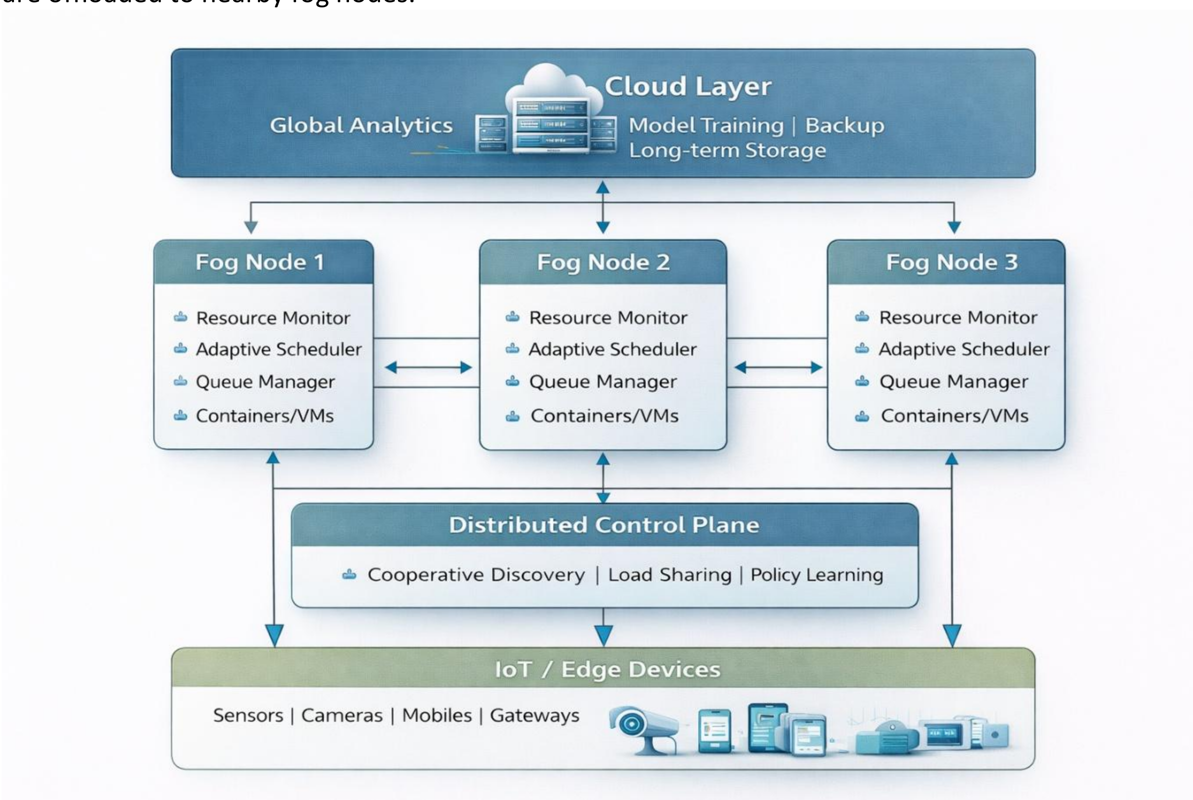

Fig. 1. System Architecture

The Fog Layer forms the core of the system and consists of multiple distributed fog nodes. Each node is equipped with a local adaptive scheduler that makes real-time decisions based on resource availability, workload conditions, and network state. Unlike centralized approaches, scheduling decisions are made locally, reducing latency and improving scalability. Additionally, fog nodes support cooperative scheduling, enabling them to share workload information and assist one another during overload conditions. The Cloud Layer acts as a backup execution environment and is invoked only when fog nodes are unable to satisfy task constraints. This selective usage minimizes cloud dependency while ensuring reliability.

➢ *Mathematical Formulation*

The scheduling problem is modeled as a multi-objective optimization problem. Let Let $T = \{t_1, t_2, \dots, t_n\}$ be the set of tasks and $F = \{f_1, f_2, \dots, f_m\}$ be the set of fog nodes. A binary decision variable x_{ij} is defined such that x_{ij} = 1 if task i is assigned to node j, and 0 otherwise.

- The total latency of a task includes transmission, computation, and queueing delays:

(a) Transmission Delay

$$T_{ij}^{tx} = \frac{d_i}{b_{ij}}$$

where $b_{ij}$is the available bandwidth between the device and the node.

(b) Computation Delay

$$T_{ij}^{comp} = \frac{w_i}{C_j}$$

(c) Queueing Delay

Let $Q_j$denote the current workload at the node j:

$$T_{ij}^{queue} = \frac{Q_j}{C_j}$$

Total Latency

$$L_{ij} = T_{ij}^{tx} + T_{ij}^{queue} + T_{ij}^{comp}$$

- Energy consumption is modeled as:

$$E_{ij} = \alpha_j w_i + \beta_j d_i$$

where
α_j= processing energy coefficient
β_j= communication energy coefficient.
Total node energy usage:

$$E_j^{used} = \sum_i x_{ij} E_{ij}$$

- Load utilization of each node is given by:

$$U_j = \frac{\sum_i x_{ij} w_i}{C_j}$$

To avoid hotspots and failures, balanced utilization across fog nodes is desired.
Load variance:

$$\sigma^2 = \frac{1}{M} \sum_{j=1}^{M} (U_j - \bar{U})^2$$

Minimizing variance improves fairness and reliability.

- Reliability decreases with overload and energy depletion. Let node reliability be:

$$R_j = e^{-\lambda_j U_j}$$

System reliability:

$$R_{sys} = \prod_{j=1}^{M} R_j$$

Maximizing this term improves fault tolerance.

The overall objective is to minimize a weighted cost function:

$$\min_{X} \; J(X)$$

$$J(X) = \lambda_1 \sum_{i,j} x_{ij} L_{ij} + \lambda_2 \sum_{i,j} x_{ij} E_{ij} + \lambda_3 \sigma^2 - \lambda_4 R_{sys}$$

where

$\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are tunable weights controlling trade-offs.

subject to constraints that ensure each task is assigned to exactly one node, that resource capacities are not exceeded, and that deadlines are satisfied. This formulation captures the trade-offs between latency, energy efficiency, and load balancing.

- *Proposed Scheduling Framework*

The proposed methodology follows an Adaptive Cooperative Multi-Objective Scheduling (ACMOS) approach designed for decentralized environments.

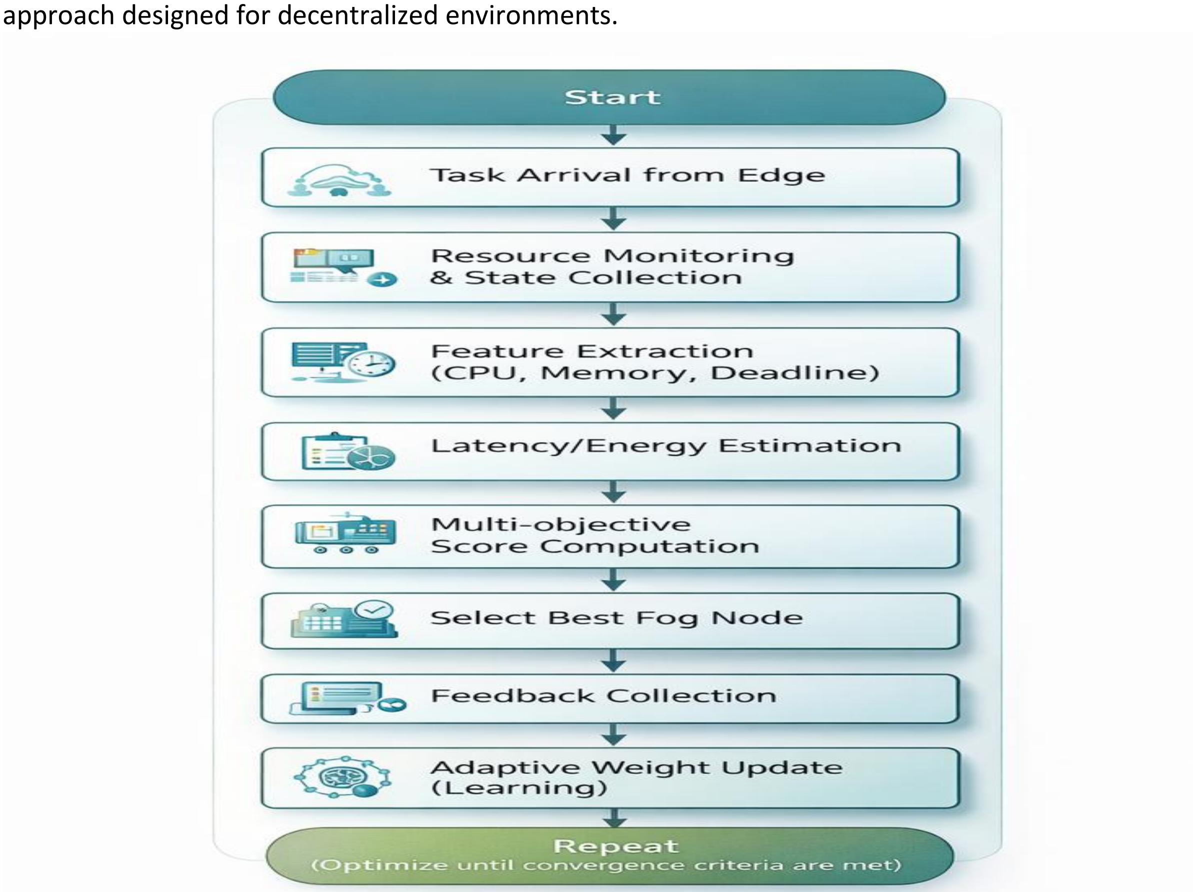

Fig. 2. Proposed Workflow

Initially, each fog node performs local scheduling, assigning tasks based on the minimum cost computed from latency, energy, and load conditions. If local execution is not feasible, the system activates a cooperative scheduling mechanism that queries neighboring fog nodes to identify a suitable execution node. This avoids unnecessary cloud communication and improves system efficiency.

If no fog node can satisfy the task constraints, the task is finally offloaded to the cloud. This results in a three-level decision strategy: local → cooperative → cloud, ensuring both efficiency and reliability.

To improve adaptability, the framework incorporates a learning-based weight adjustment mechanism. After each scheduling cycle, a reward is computed based on system performance, and weights are updated dynamically as:

$w_k^{new} = w_k^{old} + \eta R$

This enables the scheduler to continuously adapt to changing workload conditions without manual tuning.

- *Workflow Description*

The overall workflow of the proposed system can be summarized as follows: when a task arrives, the local fog node evaluates its feasibility using the cost function. If feasible, the task is executed locally. Otherwise, neighboring nodes are explored through cooperative scheduling. If no suitable node is found, the task is offloaded to the cloud. After execution, system performance is evaluated, and scheduling parameters are updated to improve future decisions.

- *Key Contributions of Methodology*

The proposed framework offers several advantages. First, decentralized scheduling reduces latency and eliminates single points of failure. Second, cooperative task sharing improves resource utilization and load balancing. Third, adaptive learning enables real-time optimization under dynamic conditions. Finally, the framework ensures scalability and reliability, making it suitable for large-scale fog-enabled IoT environments.

Conclusion

This paper presents an adaptive and efficient scheduling framework for decentralized fog computing environments to address challenges in latency, energy efficiency, and scalability in IoT systems. By distributing scheduling intelligence across fog nodes, the framework enables real-time local decision-making while eliminating dependence on centralized control. The incorporation of cooperative scheduling among neighboring nodes further improves resource utilization and reduces unnecessary reliance on cloud infrastructure.

The scheduling problem was formulated as a multi-objective optimization model, and an adaptive cost-based mechanism was introduced to dynamically balance latency, energy consumption, and load. The proposed hierarchical strategy—local execution, cooperative offloading, and cloud fallback—ensures efficient and reliable task processing. Overall, the framework provides a scalable and robust foundation for decentralized fog environments, with future work focusing on advanced learning-based optimization and comprehensive performance evaluation.

References

1. Abreha, H. G., Hayajneh, M., & Serhani, M. A. (2022). Federated learning in edge computing: a systematic survey. Sensors, 22(2), 450.
2. Fan, Y., Ge, J., Zhang, S., Wu, J., & Luo, B. (2023). Decentralized scheduling for concurrent tasks in mobile edge computing via deep reinforcement learning. IEEE Transactions on Mobile Computing, 23(4), 2765-2779.
3. Ibrahim, M. A., & Askar, S. (2023). An intelligent scheduling strategy in fog computing system based on multi-objective deep reinforcement learning algorithm. IEEE Access, 11, 133607-133622.
4. Wang, Z., Goudarzi, M., Gong, M., & Buyya, R. (2024). Deep reinforcement learning-based scheduling for optimizing system load and response time in edge and fog computing environments. Future Generation Computer Systems, 152, 55-69.
5. Mattia, G. P., & Beraldi, R. (2024). Online Decentralized Scheduling in Fog Computing for Smart Cities Based on Reinforcement Learning. IEEE Transactions on Cognitive Communications and Networking, 10(4), 1551-1565.
6. Liu, F., Liu, Z., Liu, X., & Zhou, H. (2025). Scalable hybrid framework for real time and non real time task scheduling in fog computing using federated reinforcement learning and PSO GA. Scientific Reports, 15(1), 38356.
7. Chhabra, G.S., Satti, S.K., Rajareddy, G.N.V. et al. Time-and-Traffic-aware collaborative task offloading with service caching-replacement in cloud-assisted mobile edge computing. Cluster Comput 28, 900 (2025). https://doi.org/10.1007/s10586-025-05629-x
8. Chhabra, G. S., Rajareddy, G. N., Mahapatra, A., Mangalampalli, S. S., Sahoo, K. S., Sethi, D., & Mishra, K. (2025). Deep Learning-centric Task Offloading in IoT-Fog-Cloud Continuum: A State-of-the-Art Review, Open Research Issues and Future Directions. IEEE Access.