Learning-Based Service Selection for Latency Optimization in Heterogeneous Fog Computing

Haoshen Zhai

{t330033048@mail.bnbu.edu.cn}

Faculty of Science and Technology, Beijing Normal-Hong Kong Baptist University, Zhuhai, Guangdong, 519000, China

Abstract. Edge and fog computing has become an important sample for supporting the latency sensitive application like the Internet of Things (IoT) today. However, heterogeneous network conditions and dynamic workload make the static service selections have unexpected performance in practical deployments. This study researches the self-adaptive service selection problem. It describes the problem as a Multi-Armed-Bandit (MAB) decision process and implements an Upper-Confidence-Bound (UCB) strategy in the Yet Another Fog Simulator (YAFS) simulated framework and compares with the Round Robin baseline under the heterogeneous network conditions. The result shows that the UCB-based method achieves lower long-term latency and better behavior in service selection. These findings are consistent with the recent research in routing and learning-driven service placement in fog and edge environments.

Keywords: Fog Computing, Edge Computing, Service Selection, Multi-Armed Bandit, Latency Optimization

1 Introduction

Due to the growth of sensitive latency applications like Internet of Things (IoT) and intelligent transportation systems, the efficient distribution of resources in edge and fog computing becomes significantly important. Edge and fog paradigms reduce network congestion and promote response time by bringing the computing assignment closer to end users [1]. This is critical for the requirement of meeting strict Quality of Service. However, the inherent heterogeneity and dynamic conditions of edge environments bring significant challenges for traditional static resource allocation techniques. Hence, the mechanisms of decision-making that can adapt to changing network and workload conditions are essential for optimizing the performance in these systems.

Recently, several studies have explored the adaptive techniques of edge resource arrangement under dynamic conditions. Kim et al. investigated distributed task offloading and resource allocation strategies to minimize latency in mobile edge computing networks, which showed

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365145

that dynamic decision-making can reduce end-to-end latency compared with static allocation methods, especially under high traffic load conditions [2]. Khan et al. summarized that learning-driven methods consistently achieve lower latency and better task scheduling performance than rule-based strategies in dynamic environments [3]. Benaboura et al. proposed a deep reinforcement learning–based task offloading scheme for IoT and cloud systems [4]. The experiment indicates that the proposed method reduced the average task latency by approximately 25%-35% [4]. Although these works have presented the potential of adaptive and learning-driven solutions, many existing methods only rely on simplified models or lack the systematic evaluation of end-to-end latency performance in a realistic simulation environment.

The research regards the edge service selection problem as a Multi-Armed Bandit (MAB) decision problem, and implements a selection strategy based on the Upper Confidence Bound (UCB) algorithm within a discrete-event fog simulation framework. Using heterogeneous network scenarios and Yet Another Fog Simulator (YAFS)-generated workloads, the study compares the UCB policy with the polling baseline policy to evaluate the latency performance. The goal of this study is to prove that the selection strategy based on online learning can effectively adapt to dynamic edge conditions and improve the delay results

2 System Model and Methodology

2.1 Edge–Fog Computing Scenario Description

The graph-based edge-fog system model is very popular in the literature on edge/fog computing, and represents a commonly accepted system abstraction [5, 6]. Realistic deployment conditions are represented by modeling the network topology as a graph in which nodes represent computing devices, and arcs model communication connections among them, with each arc being attributed by properties like bandwidth or propagation delay in order to account for a heterogeneous transmission behavior.

To simulate realistic fog deployment scenarios, this study uses a fog node selection strategy based on network centrality measures. Network centralities are often utilized for approximating the realistic fog deployment scenario, where fog nodes are normally located at highly connected sites of the network [3]. This study implements the entire system using YAFS that supports event-driven simulation with a fine-grained control to set up the topology, workload generation, and service routing decision [7].

2.2 Task Generation and Workload Modeling

The study chooses deterministic workload model which is used in generating tasks. It makes the experiment results repeatable, which also provides an opportunity for comparing various service selection schemes fairly. The research selects some fixed proportions of nodes from the network as task generators; these nodes will periodically produce service requests and then deliver them to a fog node for processing.

In the experiment setting, each task has a fixed instruction length and a fixed data size. The values of these two attributes are fixed in all experiments. With the same task features, the influence of workload randomness is mitigated.

This type of workload modeling is common in simulation-based works. This allows us to decouple the impact of scheduling and routing policies from that due to different task arrival patterns [2]. Thus, this research conclude that performance differences are due to the services selections mechanisms.

2.3 Service Selection as a Multi-Armed Bandit Problem

In this research, the service selection process is formulated as an MAB problem. Each available fog service instance is treated as an arm. Selecting a service corresponds to pulling one arm [8]. After the task is completed, the observed end-to-end latency is used as feedback.

The main advantage of MAB in this experiment is its capacity to support online decision-making without requiring prior knowledge of system performance. In addition, MAB is model-free and lightweight, which makes it suitable for YAFS. These advantages enable MAB to become an appropriate choice for the adaptive service selection in fog computing systems with limited prior information [1].

The paper chooses latency as the primary performance metric since it has a direct impact on Quality of Service for time-critical edge applications [4]. The smaller the value of latency is, the better service it provides. Thus, the research has a smaller latency, which means a larger reward in the learning process.

2.4 Integration of Learning-Based Selection into YAFS

The research implements the learning-based service selection algorithm in YAFS as an extension of the default service selection function. Based on the YAFS framework, the study additionally enables a dynamic routing decision.

The selection module at every routing decision makes a choice of a target fog node according to the current policy. Upon receiving the task response, the observed delay is recorded. This value will be later used to update the inner-state of the learning algorithm. In this way, the system slowly adapts to heterogeneous network conditions.

This kind of adaptability has been frequently observed in learning-based resource management works [9]. By contrast, the Round Robin policy uses the original YAFS selection scheme. In this way, the research is able to make an unbiased and controlled comparison of the adaptive vs. the non-adaptive approaches.

2.5 Experimental Configuration and Evaluation Metrics

All the experiments are run with the same simulation setup. The length of simulations, topology, workload parameters, and service deployment settings, respectively. The random seeds are set as a constant for reducing randomness.

The main evaluation metric is end-to-end latency. It is defined by the difference in time between when a task is emitted and when it is received. Besides, the study computes the cumulative latency and cumulative regret curves, based on the simulation log. The two measures are commonly adopted for assessing a bandit-based algorithm’s long-term performance and its learning behavior [10].

Taken together, they give an idea of the model's immediate performance as well as its longer-term learning behavior.

3 Experimental Results and Analysis

3.1 Experimental Setup Overview

The experiments are evaluated with YAFS simulator. The study aims at evaluating learning-based service selection under a fog computing scenario. Heterogeneous network topology is built up according to graph topologies derived from real-world ones. Transmission delay of fog node and link varies.

The experiments are set to be on the same topology, workload setting, and simulation time length. This way the experiments are able to compare different service selection strategies fairly.

The study evaluates the two strategies. The first is the Round Robin (RR) baseline. RR distributes incoming service requests evenly among fog nodes and does not use past performance information. The second strategy is Upper Confidence Bound (UCB). UCB models service selection as a multi-armed bandit problem. Each fog service is treated as an arm, and observed latency is used as feedback.

At each decision step t, the UCB strategy selects the service i that maximizes the following index (1):

$$UCB_i(t) = \bar{x}_i(t)t\sqrt{\frac{2lnt}{N_i(t)}} \tag{1}$$

In this formula, $UCB_i(t)$ denotes the confidence index of service i at decision step t. t means the times of service selection decisions made so far. $\bar{x}_i(t)$ means the average reward obtained from service i up to time t. $N_i(t)$ is the number of times service i has been selected. The term $ln\, t$ represents the natural logarithm of t. This experiment converts latency into a reward by taking its inverse. Therefore, lower latency means higher reward.

Simulation outputs are collected as event-level logs. These logs are later used to compute latency-related metrics and cumulative performance measures.

3.2 End-to-End Latency Performance

Fig. 1 shows the time-series behavior of average end-to-end latency for RR and UCB strategies.

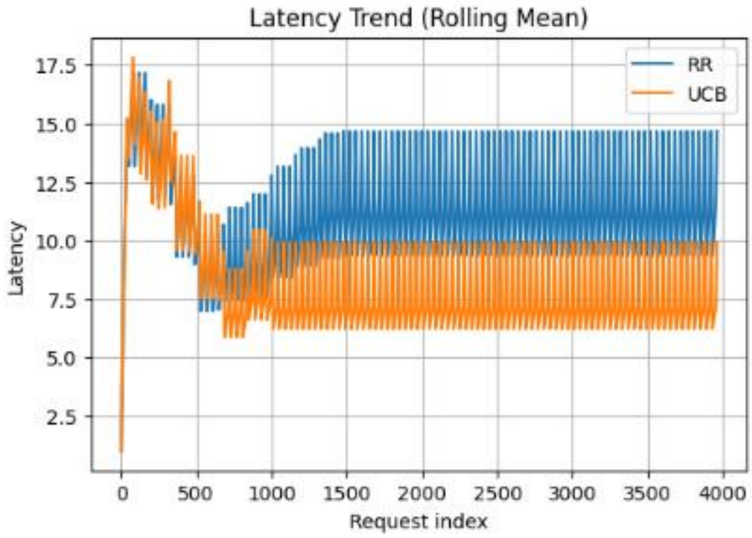

Fig. 1. Latency Performance Comparison: UCB vs. RR (Photo/Picture credit: Original).

The x-axis is the request index and the y-axis shows the end-to-end latency in simulation time units. The curves are rolling mean latency over a fixed window size.

The RR policy exhibits an almost steady latency trajectory over time. This is because it is not adaptive. On the contrary, the UCB strategy exhibits greater variation in latency at the beginning of a simulation and that over time latency is more consistent with continued learning.

Despite of adapting to heterogeneous links and fog nodes' performances, the total latency reduction with respect to RR is rather small. This finding indicates that learning based approaches are able to improve decisions' quality, but they are restricted to the network structure, and a small difference of performances between fog nodes. Similar findings were presented also for milder heterogeneous environments [2].

3.3 Cumulative Latency Comparison

To understand the longer-term behavior of a system the experiment looks at cumulative latency curves such as those given in Fig. 2. It shows the cumulative latency of RR and UCB and their differences.

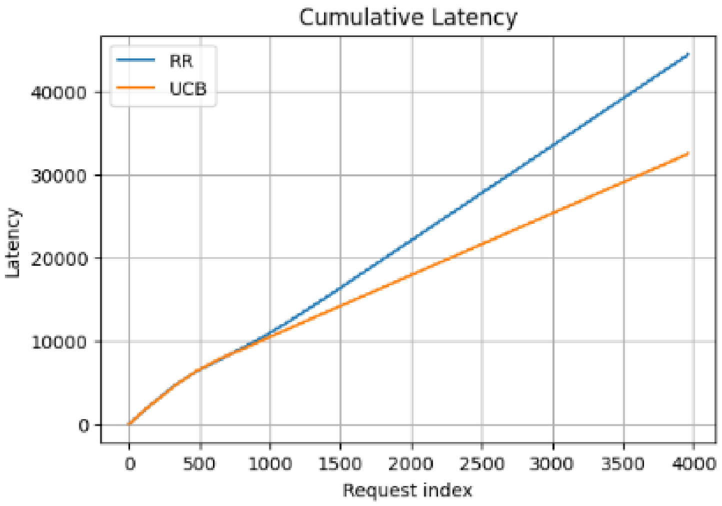

Fig. 2. Comparative Analysis of Cumulative End-to-End Latency (Photo/Picture credit: Original).

The x-axis shows the selected fog node IDs, while the y-axis is the selection fraction (0–1). The blue line is Round Robin (RR), which distributes requests almost evenly. The orange line is UCB which concentrates most requests on a few nodes.

The accumulated delay of UCB grows slower compared to RR. This shows a higher long-term performance but note how little the difference is in the distance of both curves which means the learning gain accrues only slowly instead of being immediate.

This behavior is in line with the theory for bandit algorithms: early exploration can compensate a short-term gain, especially if they have similar rewards [8]. This suggests that UCB should be used in longer running systems.

3.4 Cumulative Regret Analysis

Cumulative regret is used to measure the learning efficiency of the UCB strategy. Fig. 3 presents the cumulative regret of UCB and RR algorithms over time.

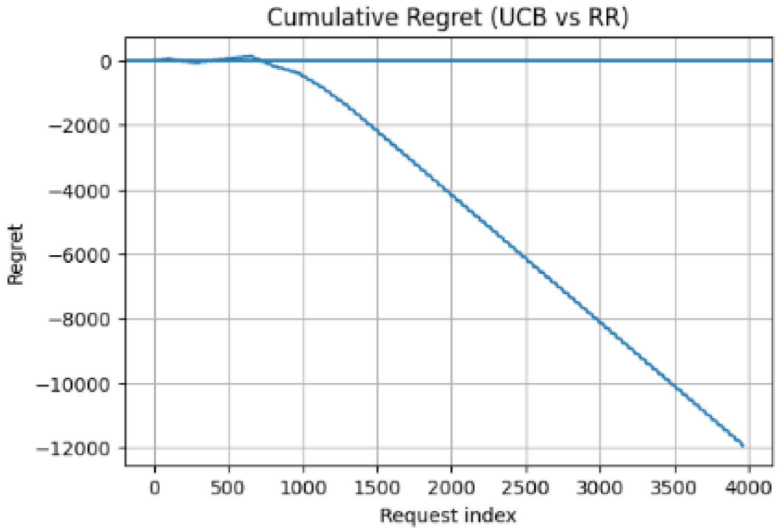

Fig. 3. Convergence of Cumulative Regret in Adaptive Service Selection (Photo/Picture credit: Original).

The x-axis is the request index, and the y-axis is the cumulative value computed from per-request latency feedback. A decreasing curve indicates that UCB achieves lower cumulative latency than RR under the same workload.

Early in the simulation, regret grows rapidly. That is expected as this corresponds to the period where the experiments explore and thus make many non-optimal choices. Over time, the regret increases at a slower pace. It means that slow learning towards good services happens.

However, the regret doesn't entirely converge to a flat curve. That is an indication of the partial reward gap and environment dynamics preventing full convergence. This behavior has also been seen for fog computing use cases where there was little performance difference between the services [4].

3.5 Discussion and Limitations

The result indicates that the UCB based selection may be better than RR in a heterogeneous fog environment. However, there still exists a limitation on the performance gain. The main cause for this issue might lie in the fact that the simulated network has only a mildly heterogeneous characteristic. It weakens the gap among different service arms while making it harder to learn.

The second drawback is using latency as the sole reward signal. There are other considerations, like energy cost or task drop rate, which are not to be considered here. Past work has suggested that multi-objective learning methods might yield better separation of performance [3].

These findings indicate that learning-based service selection should be applied in environments with higher complexity and stronger heterogeneity to fully exploit its advantages.

For the future work, it can solve these limitations from multiple aspects. Exploring more complex and highly dynamic scenarios can better highlight the benefits of learning-based service selection. Nevertheless, taking the multi-objective reward formulations into consideration can provide a more realistic system performance evaluation. Finally, by extending the current method to more advanced online learning algorithms, the adaption of fog computing system can be further improved.

4. Conclusion

This study investigates the self-adaptive service selection problem under the fog computing environment through modelling the process as MAB problem. Within the YAFS simulation framework, the UCB-based strategy was implemented and assessed under the heterogeneous network condition. The experiment result indicates that compared with non-adaptive method, leaning-based strategies can gradually promote the service selection decision and achieve lower long-term latency.

By systematic analysis, the research shows that UCB is able to balance the exploration and exploitation effectively when selecting the fog services, which enables the system to adapt the heterogeneous link delays and node performance. The cumulative result demonstrates that leaning-based selection yields higher long-term efficiency. The observation is consistent with recent research which shows better performance of adaptive and learning-driven approaches than the static strategies in fog and edge computing environment.

However, some limitations are found in this research. Firstly, the simulated network only performs mild heterogeneity. It reduces the performance gap among the service instances and limits the learning strengths of UCB. Second, without considering the energy consumption, task failure probability and system reliability, latency is used as the only reward. These limitations restrict the comprehensiveness of assessment.

References

[1] Wang, L., Zhang, J.: Adaptive multi-armed bandit learning for task offloading in edge computing. arXiv. pp. 1–15 (2023)

[2] Kim, M., Jang, J., Choi, Y., Yang, H. J.: Distributed task offloading and resource allocation for latency minimization in mobile edge computing networks. IEEE Access. Vol. 12, pp. 52130–52142 (2024)

[3] Khan, F. U., Shah, I. A., Jan, S., Ahmad, S., Whangbo, T.: Machine learning–based resource management in fog computing: A systematic literature review. Sensors. Vol. 25, No. 3, pp. 687–702 (2025)

[4] Benaboura, A., Bechar, R., Kadri, W., Ho, T. D., Pan, Z., Sahmoud, S.: Latency-aware and energy-efficient task offloading in IoT and cloud systems with DQN learning. Electronics. Vol. 14, No. 15, Art. 3090, pp. 1–18 (2025)

[5] Mach, P., Becvar, Z.: Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials. Vol. 19, No. 3, pp. 1628–1656 (2017)

[6] Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., Polakos, P. A.: A comprehensive survey on fog computing: State-of-the-art and research challenges. IEEE Communications Surveys & Tutorials. Vol. 20, No. 1, pp. 416–464 (2018)

[7] Lera, I., Guerrero, C., Juiz, C.: YAFS: A simulator for IoT scenarios in fog computing. IEEE Access. Vol. 7, pp. 91745–91758 (2019)
[8] Han, X., Xu, X.: Multi-armed bandit algorithm for online offloading and scheduling in edge computing environment. Proc. International Conference on Computing, Networking and Communications (ICNC). IEEE. pp. 81–87 (2024)
[9] Lattimore, T., Szepesvári, C.: Bandit algorithms. Cambridge University Press. pp. 1–476 (2020)
[10] Xu, Y., Liu, X., Liu, A.: Adaptive task scheduling with reinforcement learning in fog computing. Future Generation Computer Systems. Vol. 115, pp. 1–12 (2021)