

# Toward Reinforcement Learning-based Framework for Workflow Migration: Position Paper

Nour El Houda Boubaker<sup>1</sup>, Karim Zarour<sup>1</sup>, Nawal Guermouche<sup>2</sup> and Djamel Benmerzoug<sup>1</sup>

<sup>1</sup>Constantine2 - Abdelhamid Mehri University, LIRE Laboratory, Constantine, Algeria

<sup>2</sup>LAAS-CNRS, University of Toulouse, INSA

## Abstract

The concept of service migration holds significant importance within the realm of Fog-Edge computing, particularly in scenarios where mobile users are in constant motion, transitioning between various Access Points (APs). While this dynamic mobility is a fundamental characteristic of modern networking environments, it introduces the challenge of the frequent migration of services. The latter can potentially lead to a degradation in the Quality of Service (QoS) experienced by users. This paper addresses this critical issue by presenting a methodical Reinforcement Learning framework for necessary workflow migration in Fog-Cloud Computing. Firstly, we examine the literature solutions and then, we introduce our Markov Decision Model (MDP) for workflow migration.

## Keywords

Workflow Migration, Fog, Cloud, Reinforcement Learning

## 1. Introduction

User mobility in Fog and Edge environments refers to scenarios where users are constantly moving within the network, such as in a smart city or Internet of Things (IoT) deployment. As users move, their proximity to Fog nodes may change, and it becomes necessary to migrate services to Fog nodes that are closer to the users to provide seamless connectivity and optimal performance [1, 2].

However, it is important to highlight that frequent migration could result in added migration expenses, including delays and increased energy usage. Hence, there exists a need to minimize the frequency of migrations while still adhering to users' Quality of Service (QoS) requirements, such as diminishing latency as perceived by users [3]. Moreover, the regions may have varying levels of resources (computational capacity, memory, etc.) and network bandwidth [4]. These differences in resource capacities and network capabilities need to be considered when making decisions about where to migrate services to ensure optimal performance and resource utilization.

Recently, researchers have shown an increasing interest in leveraging artificial intelligence techniques to propose intelligent solutions for service migration problems in Fog and Edge


---

TACC 2023: Tunisian-Algerian Joint Conference on Applied Computing, November 6 - 8, Sousse, Tunisia

✉ [nour.boubaker@univ-constantine2.dz](mailto:nour.boubaker@univ-constantine2.dz) (N. E. H. Boubaker); [zarour.karim@univ-constantine2.dz](mailto:zarour.karim@univ-constantine2.dz) (K. Zarour); [nawal.guermouche@laas.fr](mailto:nawal.guermouche@laas.fr) (N. Guermouche); [djamel.benmerzoug@univ-constantine2.dz](mailto:djamel.benmerzoug@univ-constantine2.dz) (D. Benmerzoug)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

environments [5, 6]. These solutions aim to identify an optimal migration policy based on user mobility patterns. Our approach will employ a specific machine-learning paradigm called reinforcement learning. This letter is particularly well-suited for addressing the challenges posed by complex environments that require adaptability in response to contextual factors. Through this technique, we can develop a migration solution that dynamically adjusts the placement strategy by considering the varying performance of resources and bandwidth links. In contrast to existing solutions, our approach will primarily emphasize minimizing the frequency of workflow migrations within such heterogeneous environment characteristics to maintain a trade-off between QoS and migration costs.

The rest of the paper is organized as follows: In Section 2, we discuss the problem statement, highlighting certain limitations in existing migration solutions that motivated our approach. Section 3 provides an overview of the system modeling. In Section 4, we explain our MDP modeling and RL framework.

## 2. Problem Statement

Several works were proposed to solve the service migration problem in Fog-Edge Computing [7]. For instance, [5] proposed a deep Q-learning algorithm to solve the task migration problem without knowing users' mobility patterns. Wang and al. [6] proposed a Double Deep Q-Learning (DDQN) for computation offloading and migration framework in vehicular networks, which considers time-varying channel states and stochastically arriving computation tasks. [8], considered a centralized controller for service allocation and migration. [9] designed a DRL approach to deploying an optimal migration policy in order to improve user QoS in Mobile Edge Computing (MEC). The approach consists of migrating data to another eNodeB (eNB) depending on the user position and the current state of the network. Djemai et al. [10] presented a probabilistic mobility-based Genetic Algorithm (MGA) and mobility greedy heuristic (MGH) for an efficient services migration in the Fog environment that minimizes the infrastructure energy consumption and applications delay violations over time. Huang and al. [11] proposed an intelligent task migration scheme in MEC using the Q-learning technique. The authors aim to minimize the overall service time. [12] focused on the problem of service migration where users move between multiple edge nodes and propose a service migration strategy algorithm (SMSMA) based on multi-attribute MDP to make migration decisions.

In many existing research studies [8, 10, 9, 12], the primary focus revolves around exploring migration strategies that relocate services or workflows whenever users change locations. However, it's essential to recognize that such migration strategies can introduce computational resource overhead, higher communication costs, and longer migration times. These unnecessary migrations can deplete resources and disrupt service execution, potentially leading to suboptimal performance and operational inefficiencies.

Conversely, studies aiming to reduce migration frequency [5, 6, 11] often focus on scenarios involving a single service migration in relatively homogeneous environments. In real-world situations, regions can significantly differ in characteristics, especially regarding resource capacities and network conditions. This heterogeneity adds complexity, requiring a more nuanced approach to service migration management.

### 3. System Model Overview

In our study, depicted in Fig. 4.2, we examine a typical industrial scenario where a robot traverses geographic regions covered by Fog servers. These servers connect to Access Points (APs) through wireless links. Initially, the robot assigns computation tasks to Fog resources in the first region. However, as the user moves, the system must make informed decisions about task migration. These decisions consider factors like resource performance and network conditions. To accommodate user mobility, the system operates in time slots, with the timeline represented by  $t \in T = \{0, 1, 2, \dots, T\}$ . The duration of each time slot  $t_t$  is  $\delta$  (e.g., 15 minutes).

The robot's cyber workflow consists of tasks with dependencies. The robot's physical component continuously sends data gathered from various sensors, like images, to this workflow. This data is processed and manipulated within the cyber workflow. After processing, the results are returned to the physical component. The cyber workflow is formally represented as a directed acyclic graph (DAG), denoted as  $G = (Ts, E)$ . Here,  $Ts$  is the set of tasks, and  $E$  represents the dependencies or constraints between task pairs. Each task in this model has attributes like size and computational requirements.

Fog resources are distributed in each region, forming a network of interconnected nodes. These resources are linked together through wireless links, which can vary in characteristics from one resource to another. Furthermore, the resources are characterized by a set of attributes such as computational capacities. Our primary goal is to develop a decision-making algorithm that efficiently minimizes both the overall delay and energy consumption associated with processing in the system. This reduction includes offloading processing, execution processing, and migration processing. It encompasses the time from when tasks are offloaded to the resources of the initial region to the time the final task in the workflow completes its execution in the last region.

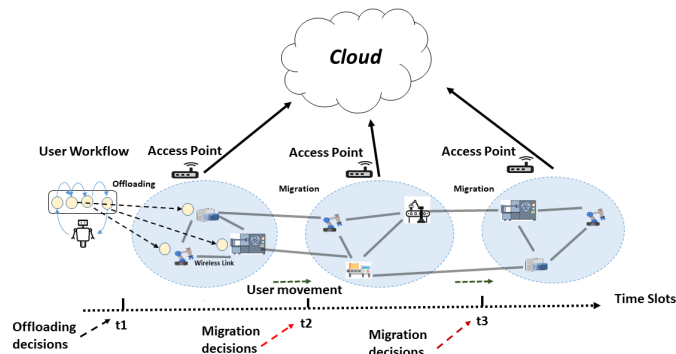


Figure 1: Overview of the system model

### 4. Workflow migration-based RL Methodology

In contrast to other fields of Machine Learning, reinforcement learning relies on continuous interaction with the environment, where the agent learns through feedback in the form of

values assessing its actions.

#### 4.1. MDP Model

The workflow migration problem is formalized as a Markov Decision Process (MDP), denoted as  $MDP = \langle S, A, R, P \rangle$  [13], where  $S$  represents the state space,  $A$  is the action space encompassing all possible actions at each state,  $R$  is the reward function valuing state-action pairs, and  $P$  determines the probability of transitioning between states when specific actions are taken.  $S$ ,  $A$ , and  $R$  are represented as follows:

1. **State Space:** The state space is defined by several variables that collectively represent the system state. These variables include the current time slot ( $t_t$ ), the information of the current task ( $\tau_i$ ) that needs to be allocated, the information of resources at the current region ( $R^t$ ), and the action taken for the task ( $A_i^{t-1}$ ) in the previous time slot. The state can be denoted as  $S_i^t = \{t_t, \tau_i, R^t, A_{i(t>1)}^{t-1}\}$ . The total number of states in one episode is equal to  $N \times T$ , where  $N$  represents the number of tasks, and  $T$  denotes the number of time slots.
2. **Action Space:** In each time slot, an action  $A_i^t$  must be taken for each task  $\tau_i$ . The action  $A_i^t \in \{a_{0(t>1)}, a_1\}$  consists of two options:  $a_0$  denotes no migration decision, while  $a_1$  involves migrating the task to the current region by selecting a suitable resource.
3. **Reward:** It represents delay and energy consumption associated with the execution of action  $A_i^t$  within the context of state  $S_i^t$ .

#### 4.2. Reinforcement Learning Framework

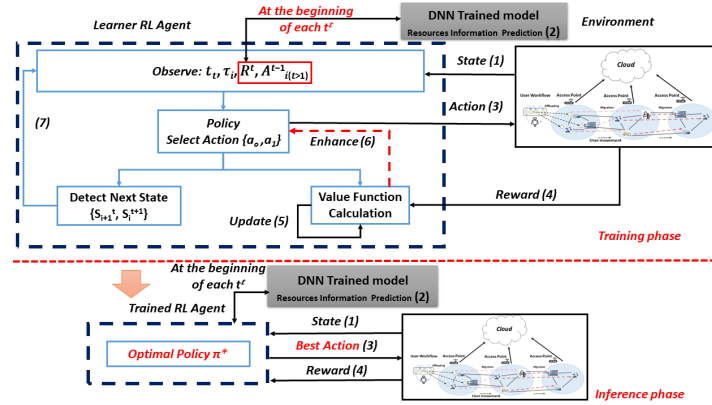
In RL, an agent interacts with an environment by taking actions, receiving rewards or penalties based on those actions, and using this feedback to improve its decision-making process. The agent's actions are guided by a policy, which determines the mapping from system states to actions [13]. The ultimate objective of the agent is to ascertain an optimal policy, denoted as  $\pi^*$ , which effectively maps a state  $s_n$  to a probability distribution over possible actions  $a_n$  and is represented as follows:

$$\pi^* : S \rightarrow P(A) \quad (1)$$

Figure 4.2 illustrates the workflow migration framework, consisting primarily of two core components: the RL agent and the migration environment. The primary aim of the agent revolves around enhancing its cumulative reward during system operations, a goal that is achieved by minimizing both overall delay and energy consumption. This cumulative reward, denoted as  $R_{total}$ , is defined as the summation of rewards of tasks across all time slots:

$$R_{total} = \sum_{i=1}^T \sum_{i=1}^N \gamma R(S_i^t, A_i^t) \quad (2)$$

In the initial stages, the agent operates without prior knowledge about the environment and therefore initiates exploration by taking random actions that might not yield immediate high rewards but offer valuable insights for discovering more rewarding actions over time. Subsequently, the agent shifts to exploitation, where it selects actions aimed at maximizing the expected future rewards, relying on its current understanding of the environment.



**Figure 2:** Reinforcement Learning Framework

At first, the agent receives the information related to the first state denoted  $s_1^1$ , including, the information of the first task (CPU, RAM, and DISK) requirements, as well as information about the available resources in the first region, including cloud resources (identified by id machine). The Deep Neural Network (DNN) prediction module utilizes this information, in conjunction with a timestamp to estimate the current resource utilization, specifically for CPU, RAM, and Disk. This step is crucial in minimizing resource overhead, as the agent will only consider the subset of resources that can adequately meet the requirements of the task. After the resource selection, the agent receives a reward that reflects the resulting delay and energy consumption from offloading and executing the task on the selected resource. To facilitate its decision-making process, the agent relies on a value function, specifically the State Value Function  $V(s)$  or the State-Action Value Function  $Q(s, a)$ . This function estimates the anticipated future reward that the agent can achieve by taking a specific action from the current system state. By utilizing the value function, the agent can evaluate the potential advantages of different actions and prioritize the most promising ones. Subsequently, the agent updates the value and requests the next state, which represents the information regarding the second task to be allocated to the resources in the first time slot. Once the agent has completed the task offloading process in the first time slot, it requires information about the first task, denoted as  $s_1^2$ , which includes its previous host machine. This information is necessary for the agent to make a migration decision  $a_1^2$  in the second time slot. The host machine's identifier is used by the prediction module to estimate the current resource utilization for this resource. This iterative process continues as the agent makes migration decisions for each subsequent task across all regions until decisions are made for all tasks.

The RL agent learns through multiple episodes, refining its value function from rewards and new information gathered during interactions with the environment. This iterative process enhances its decision-making, leading to the acquisition of an optimal policy. This value function

update is a fundamental aspect of RL algorithms like Q-learning and State-Action-Reward-State-Action (SARSA).

In the inference phase, the agent leverages its acquired knowledge to make decisions and take actions during interactions with its environment. This behavior reflects the wisdom accumulated from its training, guiding it to navigate and interact in alignment with the optimal migration policy  $\pi^*$ .

## 5. Conclusion And Future Works

This position paper addresses Workflow Migration in the context of Fog-Cloud Computing, especially in scenarios with varying resource capacities and bandwidth links. We began by highlighting limitations and challenges in existing literature. We then outlined our system model. We introduced an MDP model, defining its key components. Additionally, we presented an RL framework for necessary workflow migration.

In the future, we aim to develop a Deep Learning resource prediction module using Google cluster trace and Alibaba data, explore partial offloading for energy-constrained end-users, and investigate multi-agent strategies for scalability in expanding resource scenarios.

## Acknowledgment

This work was partially supported by the LABEX-TA project MeFoGL: "Méthode Formelles pour le Génie Logiciel"

## References

- [1] Z. Rejiba, X. Masip-Bruin, E. Marín-Tordera, A survey on mobility-induced service migration in the fog, edge, and related computing paradigms, *ACM Computing Surveys (CSUR)* 52 (2019) 1–33.
- [2] S. Wang, J. Xu, N. Zhang, Y. Liu, A survey on service migration in mobile edge computing, *IEEE Access* 6 (2018) 23511–23528.
- [3] G. Liu, J. Wang, Y. Tian, Z. Yang, Z. Wu, Mobility-aware dynamic service placement for edge computing, *EAI Endorsed Transactions on Internet of Things* 5 (2019) e2–e2.
- [4] S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in: *Proceedings of the 2015 workshop on mobile big data*, 2015, pp. 37–42.
- [5] C. Zhang, Z. Zheng, Task migration for mobile edge computing using deep reinforcement learning, *Future Generation Computer Systems* 96 (2019) 111–118.
- [6] H. Wang, H. Ke, G. Liu, W. Sun, Computation migration and resource allocation in heterogeneous vehicular networks: a deep reinforcement learning approach, *IEEE Access* 8 (2020) 171140–171153.
- [7] N. E. H. Boubaker, K. Zarour, N. Guermouche, D. Benmerzoug, Fog and edge service migration approaches based on machine learning techniques: A short survey (2022).

- [8] M. Zhang, H. Huang, L. Rui, G. Hui, Y. Wang, X. Qiu, A service migration method based on dynamic awareness in mobile edge computing, in: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2020, pp. 1–7.
- [9] F. De Vita, D. Bruneo, A. Puliafito, G. Nardini, A. Viridis, G. Stea, A deep reinforcement learning approach for data migration in multi-access edge computing, in: 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K), IEEE, 2018, pp. 1–8.
- [10] T. Djemai, P. Stolf, T. Monteil, J.-M. Pierson, Mobility support for energy and qos aware iot services placement in the fog, in: 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE, 2020, pp. 1–7.
- [11] S.-Z. Huang, K.-Y. Lin, C.-L. Hu, Intelligent task migration with deep qlearning in multi-access edge computing, IET Communications 16 (2022) 1290–1302.
- [12] P. Tian, G. Si, Z. An, J. Li, F. Zhou, Service migration strategy based on multi-attribute mdp in mobile edge computing, Electronics 11 (2022) 4070.
- [13] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.