A METHOD FOR OPTIMAL EDGE ORCHESTRATION ON CABLE NETWORKS USING REINFORCEMENT LEARNING

INVENTOR:

RAHIL GANDOTRA

Date: June 21, 2023

Submitter: Rahil Gandotra

Title: A method for optimal edge orchestration on cable networks using reinforcement learning

Background:

Machine Learning

Machine learning (ML) is a part of artificial intelligence that aims to develop methods that leverage data to improve performance on some set of tasks. ML belongs to a category of algorithms that enables software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to develop algorithms that can receive input data and use statistical analysis to predict an output, while updating outputs as new data becomes available. For simple tasks assigned to computers, it is possible to program algorithms instructing the machine how to execute all steps required to solve the problem at hand; no learning is needed by the computer. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. ML algorithms build a model based on sample data, known as training data, in order to make predictions or decisions. ML algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, computer vision, agriculture, driverless cars, and finance, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

ML can be divided into three broad categories of algorithms depending on the nature of feedback available to the learning system:

1. Supervised learning: The system is presented with example inputs and their desired outputs as training data, using which a mapping function is approximated to predict the output for new input data. Classification and regression algorithms fall under the category of supervised learning.
2. Unsupervised learning: No training data is provided to the system, leaving it on its own to discover patterns and find structure in its input. Clustering and association algorithms belong to the category of unsupervised learning.

3. Reinforcement learning: The system learns by interacting with a dynamic environment in which it performs a certain goal and is provided feedback using a system of reward and punishment, with the goal being to maximize its reward and minimize its penalty.

Reinforcement learning

Reinforcement learning (RL) framework is modeled on the problem of optimal control of Markov Decision Processes. The main elements of an RL framework include – (i) the agent or the learner, (ii) the environment the agent interacts with, (iii) the policy that the agent uses to take an action, and (iv) the reward signal observed by the agent upon taking an action. A typical RL agent interacts with its environment in discrete time steps. At each time *t*, the agent takes an action a_t transitioning the environment from state s_t to s_{t+1}. Based on the new state s_{t+1}, a reward r_{t+1} is determined and associated with the transition (s_t, a_t, s_{t+1}). The long-term goal of the RL agent is to learn a policy which maximizes the expected cumulative reward.

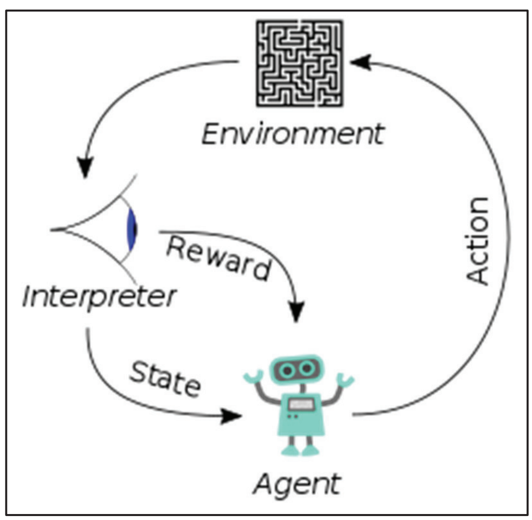

Fig. 1 Typical framing of a RL scenario

[Source: https://en.wikipedia.org/wiki/Reinforcement_learning]

RL algorithms are categorized into two broad categories: model-based and model-free. Model-based algorithms involve the agent explicitly referencing a model of the environment to choose its optimal policy. Model-based approaches include Model-Based Value Expansion (MBVE), world models, and Imagination-Augmented Agents (I2A). On the other hand, model-free algorithms are based on trial-and-error experience for constructing its optimal policy. Model-free approaches include policy optimization techniques such as policy gradients and Q-learning based techniques such as Deep Q Neural Network (DQN). In case where an accurate model of the environment is not available as part of the problem statement, model-free algorithms are more suitable than model-based algorithms. However, model-free algorithms are statistically less efficient as compared to model-based algorithms since information from the environment is combined with previous, possibly inaccurate, estimates about state values [1].

Edge computing

Edge computing is a new paradigm that enables computing at the edge of the network, closer to the source of the data. This involves integrating resources which are closer to the user in terms of geographical distance or network distance to provide computing, storage, and networking services for application deployments [2]. Computing tasks such as processing, storage, caching, and load balancing can be performed on the data sent to and from the cloud at the network edge. Some of the applications of edge computing include home automation systems, cloud gaming, XR, connected or autonomous cars, Industry 4.0 (smart industry), and smart cities. Computing at the edge allows for lower response speeds, near-real time processing, reduced load on network bandwidth, increased privacy, and reduced energy consumption.

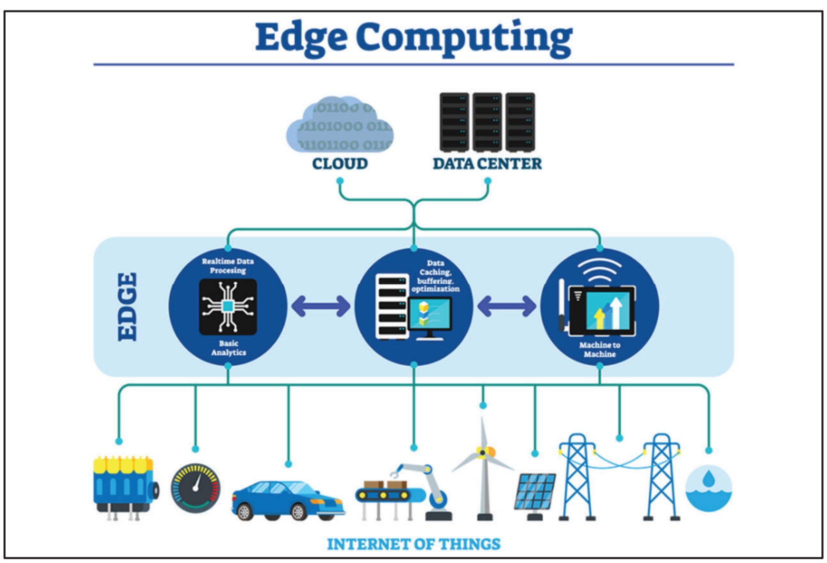

Fig. 2 Edge computing

[Source: https://innovationatwork.ieee.org/real-life-edge-computing-use-cases/]

The edge must be designed in a way such as to handle tasks efficiently, reliably, and securely. Typically, edge devices have limited computing resources as compared to the cloud, and therefore a major challenge for the network operators is to find the most efficient way to orchestrate edge network and computing resources while satisfying the needed quality of experience (QoE) requirements.

Problem statement:

Currently no method exists to utilize reinforcement learning for orchestrating computing services onto cable edge networks.

Proposed solution:

Using reinforcement learning technique for the cable edge orchestration agent to determine the optimal computing node and network path.

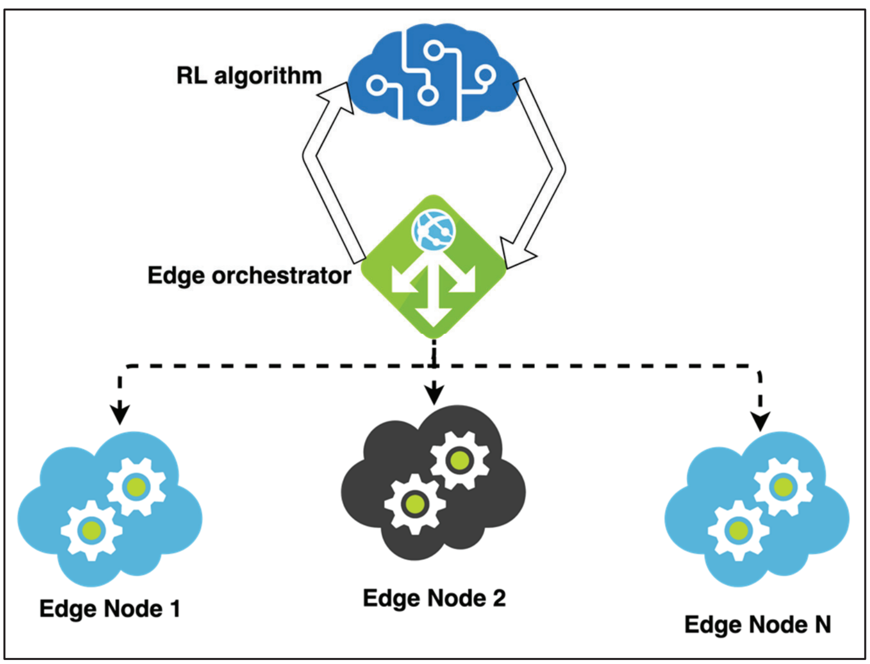

Fig. 3 RL for edge orchestration

Parameters to be considered for optimal RL-based edge orchestration –

1. Computing parameters – Memory, storage, CPUs, GPUs
2. Networking parameters – Bandwidth, latency, jitter, packet loss
3. Business parameters – Revenue, cost

We also propose a new function in the cable network core – intelligent cable edge orchestrator (ICEO). The services provided by ICEO include –

1. Optimal prediction for the placement of a computing task to the most appropriate cable edge node
2. Collecting near-real time information from the network to infer the expected reward/penalty

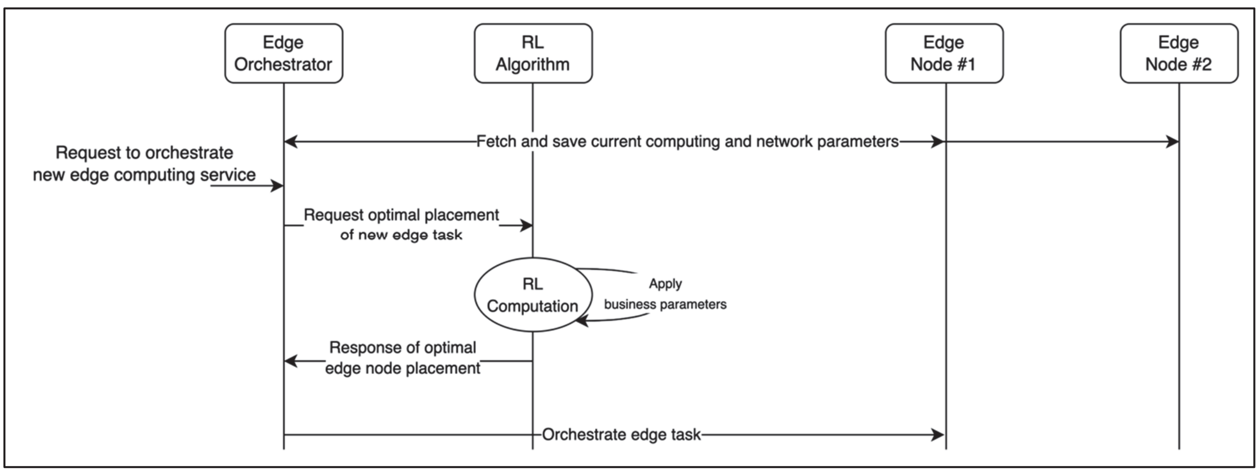

Fig. 3 Proposed interactions between Edge Orchestrator and RL Algorithm for optimal edge node placement

Benefits:

- Efficient use of available edge computing resources in cable networks
- Automated streamlining of edge application locations
- Novel application of ML for cable networks

Existing solutions:

None

Interested parties:

Edge-computing service providers, MSOs

Impact:

Efficient use of limited edge compute. Evolution of cable edge network.

References:

[1] P. Dayan and Y. Niv, "Reinforcement learning: The Good, The Bad and The Ugly," *Current opinion in neurobiology*, vol. 18, no. 2, pp. 185-196, Apr. 2008.

[2] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An Overview on Edge Computing Research," *IEEE access*, vol. 8, pp. 85714-85728, May 2020.