POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Training Reinforcement Learning agents for the computing continuum: the FIGARO framework

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Author: **Benedetta Presicci**

Student ID: 989783
Advisor: Prof. Danilo Ardagna
Co-advisors: Federica Filippini, Riccardo Cavadini
Academic Year: 2022-23

Abstract

Emerging Artificial Intelligence applications leverage all of the available computing tiers, operating seamlessly across Edge and Cloud platforms. Deploying applications at the network's periphery enables faster processing times, while cloud computing allows for increased scalability. Dealing with such a dynamic, modular, and diverse environment requires managing resources to ensure satisfaction of Quality of Service constraints and the minimization of costs. A novel framework based on Reinforcement Learning was proposed to handle the variability in a runtime computing continuum scenario: FIGARO (reinForcement learnInG mAnagement acRoss computing cOntinuum). An initial offline training period has been developed to reduce the training time where the agent learns from the insights provided by a design-time tool. However, the first version of FIGARO was limited to applications including up to two components. This work tackles this problem by generalizing the state space definition and streamlining the preliminary offline training.

Keywords: Artificial Intelligence, Reinforcement Learning, computing continuum, Optimal allocation

Abstract in lingua italiana

Le emergenti applicazioni di Intelligenza Artificiale sfruttano tutti i tier di calcolo disponibili, operando in modo fluido tra piattaforme Edge e Cloud. La distribuzione delle applicazioni ai margini della rete consente tempi di elaborazione più rapidi, mentre il cloud computing permette una maggiore scalabilità. Organizzare un sistema così dinamico, modulare e diversificato richiede una gestione delle risorse che garantisca la soddisfazione dei vincoli di Qualità del Servizio e la minimizzazione dei costi. È stato proposto un nuovo framework basato sull'Apprendimento per Rinforzo per gestire le variazioni in uno scenario di computing continuum in tempo reale: FIGARO (reinForcement learnInG mAnagement acRoss computing cOntinuum). È stato sviluppato un periodo iniziale di addestramento offline per ridurre il tempo di addestramento, durante il quale l'agente apprende dalle informazioni fornite da uno strumento di progettazione. Tuttavia, la prima versione di FIGARO era limitata alle applicazioni che includono fino a due componenti. Questo lavoro di tesi affronta il problema generalizzando la definizione dello spazio degli stati e semplificando l'addestramento offline preliminare.

Parole chiave: Intelligenza Artificiale, Apprendimento per Rinforzo, computing continuum, Allocazione Ottima

Contents

1 | Introduction

Machines created by humans can already handle various labor-intensive tasks. However, since humans are driven by the desire for better productivity, they have been attempting to give machines human-like intelligence, which is the main goal of Artificial Intelligence (AI). AI research, spanning over 65 years, has achieved remarkable advancements in theory and practical applications ([1], [2]).
There are many definitions of Artificial Intelligence. Marvin Minky, one of the pioneers of AI, defined it as enabling machines to do things that require human intelligence. Although the descriptions of AI are various, its core is widely believed to be the research theories, methods, technologies, and applications for simulating, extending, and expanding human intelligence [3]. Nowadays, the concept of AI has an increasingly profound impact on human life.
The research fields of AI include systems and engineering, brain science, healthcare, computer science, and many other disciplines, such as natural language processing. For example, an AI product used today by 20 million people is Grammarly. Grammarly [4] is a writing assistant that checks for spelling, grammar, punctuation errors, and it enhances vocabulary usage. It is a sophisticated AI system that has been improved over the years, built by linguists and engineers who developed algorithms to detect patterns of good writing. The Grammarly AI system reviews every sentence and searches for appropriate replacements for errors when any are found. It operates on a freemium model, where paid tiers give users more tools beyond grammar checks, such as plagiarism checks. Brad Hoover, the company's CEO, confirmed in 2019 to TechCrunch [5] that the company was valued at more than $1 billion. AI is a smarter way for different industries to work more efficiently than traditional methods. It reduces costs and helps distribute the work uniformly while completing it faster. It is projected to experience significant growth in the next decade, as its market share is expected to show an annual growth rate (CAGR 2024-2030) of 15.83%, resulting in a market volume of $738.80 billion by 2030 [6]. The changes in the methodology of various sectors toward automation to increase the productivity of organizations and the rise in demand for artificial intelligence and machine learning-based solutions are responsible for market growth [7].

AI and cloud computing converge in automating processes, in fact Grammarly itself is a cloud-based program. With cloud computing, individuals and organizations can gain on-demand network access to a shared pool of managed and scalable IT resources, such as servers, storage, and applications [8]. The data is stored on physical servers, which are maintained by a cloud service provider. Its primary benefits include nearly limitless storage, quick deployment, easy access to information, and simplified scaling of services [9]. For example, Grammarly data is stored on servers hosted by Amazon Web Services in the US and ensures continual product availability by using native backup tools [10]. Moreover, all components that process users data operate in Grammarly's private network inside a secure cloud platform, and each Grammarly user's data is isolated from other users' data. According to a Deloitte study [11], 70% of companies obtain their AI capabilities from cloud-based software, while 65% develop AI applications using cloud services.
However, the advent of the Internet of Things (IoT) era enables millions of interconnected devices to gather vast amounts of data processed in the cloud for various purposes [12]. IDC [13] predicted up to 60 billion connected entities by 2025, generating about 80 ZB of raw data. This data overload exposes the drawbacks of the cloud approach: network saturation, unsustainable large data centers in terms of size and energy consumption, unreliable internet connections, and latency, making it unsuitable for real-time or emergency use cases. In this context, new post-cloud trends aim to move some processing closer to the data, taking advantage of the improved abilities of edge devices. Edge computing is a paradigm that performs computing at the edge of the network, meaning closer to the source of the data, near the end devices [14]. Thanks to this approach, edge computing is characterized by low latency, processing power closer to the user and real-time interactions, overcoming cloud computing limitations. Nonetheless, edge resources usually have less computing capacity than the cloud and can become a bottleneck in the computation.

The most promising paradigm is called *computing continuum* [15], where distributed applications can seamlessly run on any device from the edge to the cloud, creating a distributed computing system that is able to fulfill highly heterogeneous applications requirements. The computing continuum performs computations by distributing the workload across multiple devices in the system. Each device handles a part of the computation, and the results are merged to generate the final output. This resource allocation is based on factors like resource proximity, computational capability, and prioritizing time-sensitive tasks. Real-time responses might be sent to edge devices, while complex analytics are usually done in the cloud, depending on the task. This dynamic distribution of tasks enhances system performance and processing efficiency while reducing latency.

As an example of computing continuum, consider one of the ELASTIC use cases: the edge/fog-enabled "Remote Sensing/Advanced Driver Assistance Systems (ADAS) from infrastructure" application for the city of Florence [16]. This use case considers a distributed infrastructure of smart cameras and computing and communication networks placed along the tramway track. The objective is to be able to detect vehicle accidents beyond the capacity of the tramway sensors/radars and to prevent accidents in case of no visibility due to urban objects occulting vehicles or pedestrians. The Figure 1.1 below shows a schematic view of the compute continuum considered in the ELASTIC use case.

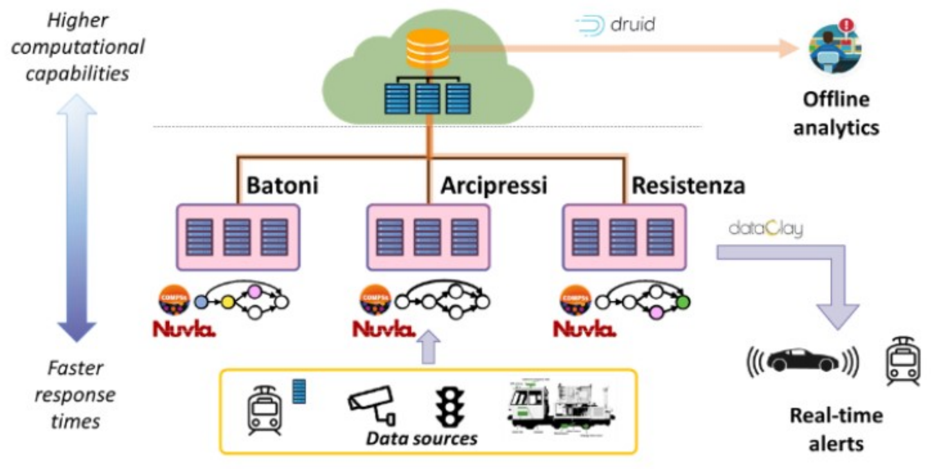

Figure 1.1: Elastic computing continuum [17].

Data sources (i.e. cameras, sensors on the tram, etc.) collect data to be sent for real-time computations. In fact, computer and data nodes can also be found in trains and stations (Batoni, Arcipressi and Resistenza) [18]. Enabled by the Edge Computing as a Service provided by Nuvla, the applications based on real-time values typically will be executed in the edge, e.g., object detection in the tramway and sending proximity alerts to the tramway cockpit ([19], [16]). In contrast, the offline analysis based on edge's historical values will run in the cloud, where the heaviest part of processing and storage can be completed ([19], [18]). A conclusion from the offline study could be the readjustment of traffic lights because of the detection of several people crossing the tramway at a particular time. In summary, the intention is to provide access to all the distributed data regardless of its source and function, without interfering between the different applications/tasks.

The design and management methodologies of these systems pose a significant challenge. The traditional methods work with client server systems, whose architecture is defined ad-hoc to solve a precise problem. While these methods remain effective for the cloud paradigm, they are inadequate for the computing continuum, where a diverse range of

architectures coexist within the same system. This challenge, typical of computing continuum systems and known as the *Resource Selection and Application Components Placement* (RS-CP) problem, is key in this work. It is usually faced at design time, however, in real-world scenarios, the expected workload and the service time distribution frequently change. Therefore, the initially planned placement may become unfeasible or excessively large depending on the workload. Hence, the authors of [20] proposed FIGARO (reinForcement learnInG mAnagement acRoss computing cOntinua), a runtime framework based on Reinforcement Learning (RL) that automatically learns to solve the RS-CP problem under varying system conditions at runtime.
The FIGARO environment embeds:

- A Simulator to compute the response times mimicking the actual response times of a cloud system [21];
- A design-time tool, SPACE4AI-D [22]. It is used to compute the response times and to represent the analytical counterpart of the Simulator, allowing a comparison. Moreover, it can also behave as the agent deployed inside the environment;
- A RL-based agent, that exploits both an initial share of knowledge and every information acquired at runtime. In particular, SPACE4AI-D is the provider of the initial knowledge. First, the FIGARO agent is trained offline to learn a policy that mimics the behavior of SPACE4AI-D. Then it is deployed at runtime.

The authors of [20] tested, validated and demonstrated that FIGARO outperforms an agent with random initial policy and a static agent adapted from the design time with an application pipeline including up to two components.

The key feature of this tool is the offline training. If the online training were to start with a random policy, the learning process would take an extremely long time and would not converge toward any meaningful solution. The policy obtained through offline training is trained using simulations of workload variations and mimics the behavior of the design-time tool. This initial policy enables significant online training to take place. Given its crucial role, it has become fundamental to enhance its efficiency.
Many different experiments are needed to advance in FIGARO development, as it is in its early stages. However, currently running offline training tests in simple scenarios is a computer-intensive and lengthy process. Moreover, due to the current definition of the state space, the policy fails to converge when the application pipeline is slightly more complex.
This work aims to contribute to the development of FIGARO, more specifically to ease the process of finding the optimal initial policy with offline training. This translates into

changes to the implementation that further encourage the agent to mimic SPACE4AI-R's behavior. Furthermore, the state space definition is changed to allow tackling the management of complex systems.

This work of thesis is organized as follows. Section 2 presents the state of the art. Section 3 introduces key concepts of Reinforcement Learning. Section 4 describes the framework at the basis of this thesis. Section 5 illustrates the contributions to FIGARO. Section 6 reports the experimental results. In Section 7 the conclusions of this work are discussed.

Bibliography

[1] Okyay Kaynak. The golden age of Artificial Intelligence. *Discover Artificial Intelligence*, 1(1), September 2021.

[2] J McCarthy, M L Minsky, N Rochester, I B M Corporation, and C E Shannon. A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE.

[3] Yuchen Jiang, Xiang Li, Hao Luo, Shen Yin, and Okyay Kaynak. Quo vadis artificial intelligence? *Discover Artificial Intelligence*, 2(1):4, March 2022.

[4] About Us | Grammarly.

[5] Ingrid Lunden. Grammarly raises $90M at over $1B+ valuation for its AI-based grammar and writing tools, October 2019.

[6] Artificial Intelligence - Global | Statista Market Forecast.

[7] Allied Market Research https://www.alliedmarketresearch.com. Artificial Intelligence Platform Market Size, Share | Forecast - 2030.

[8] Ali Sunyaev. Cloud Computing. In *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*, pages 195–236. Springer International Publishing, Cham, 2020.

[9] Diana Andreea Popescu, Noa Zilberman, and Andrew W Moore. Characterizing the impact of network latency on cloud-based applications' performance.

[10] Security at Grammarly.

[11] AI-fueled organizations.

[12] G. Massari, M. Zanella, and W. Fornaciari. Towards distributed mobile computing. In *2016 Mobile System Technologies Workshop (MST)*, pages 29–35, 2016.

[13] IoT.Business.News. IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC, July 2020.

[14] Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. An overview on edge computing research. *IEEE Access*, 8:85714–85728, 2020.

[15] Schahram Dustdar, Victor Casamayor Pujol, and Praveen Kumar Donta. On distributed computing continuum systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4092–4105, 2023.

[16] Towards an industrial computing continuum | ELASTIC.

[17] Software Infrastructure | ELASTIC.

[18] Adrián Orive, Aitor Agirre, Hong-Linh Truong, Isabel Sarachaga, and Marga Marcos. Quality of Service Aware Orchestration for Cloud&ndash;Edge Continuum Applications. *Sensors*, 22(5), 2022.

[19] Rita Sousa, Luis Nogueira, Fátima Rodrigues, and Luis Miguel Pinho. Global resource management in the elastic architecture. In *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 01–06, 2022.

[20] Federica Filippini, Riccardo Cavadini, Danilo Ardagna, Riccardo Lancellotti, Gabriele Russo Russo, Valeria Cardellini, and Andrea Lo Presti. FIGARO: reinForcement learnInG mAnagement acRoss computing cOntinua. In *3rd Workshop on Distributed Machine Learning for the Intelligence Computing Continuum at ACM/IEEE UCC 2023 (to appear)*, 2023.

[21] OMNeT++ Discrete Event Simulator.

[22] Hamta Sedghani, Federica Filippini, and Danilo Ardagna. A random greedy based design time tool for ai applications component placement and resource selection in computing continua. In *2021 IEEE International Conference on Edge Computing (EDGE)*, pages 32–40, 2021.

[23] Baudouin Herlicq, Abderaouf Khichane, and Ilhem Fajjari. Nextgenemo: an efficient provisioning of edge-native applications. pages 1924–1929, 05 2022.

[24] T. Bahreini and D. Grosu. Efficient algorithms for multi-component application placement in mobile edge computing. *IEEE Transactions on Cloud Computing*, 10(04):2550–2563, oct 2022.

[25] Federica Filippini, Hamta Sedghani, and Danilo Ardagna. SPACE4AI-R: Runtime Management Tool for AI Applications Component Placement and Resource Selection in Computing Continua. In *3rd Workshop on Distributed Machine Learning for the Intelligence Computing Continuum at ACM/IEEE UCC 2023 (to appear)*, 2023.

[26] Thiago Pereira da Silva, Aluizio Rocha Neto, Thais Vasconcelos Batista, Flávia C. Delicato, Paulo F. Pires, and Frederico Lopes. Online machine learning for autoscaling in the edge computing. *Pervasive and Mobile Computing*, 87:101722, December 2022.

[27] Yeting Guo, Fang Liu, Nong Xiao, Zhaogeng Li, Zhiping Cai, Guoming Tang, and Ning Liu. Para: Performability-aware resource allocation on the edges for cloud-native services. *International Journal of Intelligent Systems*, 37, 07 2022.

[28] Xun Shao, Go Hasegawa, Mianxiong Dong, Zhi Liu, Hiroshi Masui, and Yusheng Ji. An online orchestration mechanism for general-purpose edge computing. *IEEE Transactions on Services Computing*, 16(2):927–940, 2023.

[29] Michael Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*, volume 3. 01 2010.

[30] Qianlin Liang, Walid A. Hanafy, Ahmed Ali-Eldin, and Prashant Shenoy. Model-driven cluster resource management for ai workloads in edge clouds. *ACM Trans. Auton. Adapt. Syst.*, 18(1), mar 2023.

[31] Bhuvan Urgaonkar, Giovanni Pacifici, Prashant Shenoy, Mike Spreitzer, and Asser Tantawi. An analytical model for multi-tier internet services and its applications. *SIGMETRICS Perform. Eval. Rev.*, 33(1):291–302, jun 2005.

[32] Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, 35(3):713–728, 2005.

[33] Ida Nurcahyani and Jeong Woo Lee. Role of machine learning in resource allocation strategy over vehicular networks: A survey. *Sensors*, 21(19), 2021.

[34] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Fred Juang. Deep reinforcement learning based resource allocation for v2v communications. *IEEE Transactions on Vehicular Technology*, 68(4):3163–3173, April 2019.

[35] Zhengwei Lyu, Ying Wang, Man Liu, and Yuanbin Chen. Service-driven resource management in vehicular networks based on deep reinforcement learning. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6, 2020.

[36] Shuiguang Deng, Zhengzhe Xiang, Peng Zhao, Javid Taheri, Honghao Gao, Jianwei Yin, and Albert Y. Zomaya. Dynamical resource allocation in edge for trustable internet-of-things systems: A reinforcement learning method. *IEEE Transactions on Industrial Informatics*, 16(9):6103–6113, Sep. 2020.

[37] Sihua Wang, Mingzhe Chen, Xuanlin Liu, Changchuan Yin, Shuguang Cui, and H. Vincent Poor. A machine learning approach for task and resource allocation in mobile-edge computing-based networks. *IEEE Internet of Things Journal*, 8(3):1358–1372, Feb 2021.

[38] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[39] Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian Q-Learning.

[40] Alessandro Lazaric. Transfer in Reinforcement Learning: A Framework and a Survey. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[41] Paulo Rauber, Avinash Ummadisingu, Filipe Mutz, and Jürgen Schmidhuber. Reinforcement Learning in Sparse-Reward Environments With Hindsight Policy Gradients. *Neural Computation*, 33(6), May 2021.

[42] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[43] Replay Buffers | TensorFlow Agents.

[44] Edward D Lazowska, John Zahorjan, G Scott Graham, and Kenneth C Sevcik. *Quantitative system performance: computer system analysis using queueing network models*. Prentice-Hall, Inc., 1984.

[45] Fabio Lavezzo Andrea De Bettin. Figaro: reinforcement learning management across computing continua, 2022.

[46] Hyperopt Documentation.

List of Figures

List of Tables