Dynamic Hardware Customisation for Mobile Users in FPGA-accelerated Edge Infrastructures

Diogo Gonçalves[1], **Luiz Bittencourt**[1], **Edmundo Madeira**[1]

[1]Institute of Computing, University of Campinas, Brazil

diogomg@lrc.ic.unicamp.br, {bit, edmundo}@ic.unicamp.br

Abstract. *User mobility support is relevant for smart applications in edge network infrastructures. Machine Learning-based solutions have been taking advantage of hardware customisation to improve their performance. In this context, attaching Field Programmable Gate Array (FPGA) technology into edge nodes could offer tailored hardware for ML applications at the edge. This work proposes hardware customisation at the edge network to improve the performance of ML-based solutions. In that scenario, mobile users could configure the hardware and apply it to different edge nodes. Mobile users can use customised hardware to execute applications at the edge in a performance-enhanced environment by carrying hardware settings.*

1. Introduction

Smart Cities solutions aim to improve the quality of life of citizens by minimising urban problems. However, integrating the physical world and decision-making systems is still challenging. Machine learning (ML) algorithms and Cloud Computing infrastructures have been playing a successful role as smart city supporters. Among various smart city solutions, intelligent transportation systems which avoid traffic jams or connected video surveillance systems that improve the city's security are some examples. In that context, data collected from users are processed in cloud data centres. In those centralised servers, ML algorithms act as decision-makers. However, due to the high volume of data, some solutions might face a bottleneck performance in terms of latency, throughput, or both.

Historically, computing power has been improved by increasing the number of transistors and threads. However, in recent years, domain specialisation has received significant attention. Hardware customised for a particular application avoids unnecessary resources and focuses on specific characteristics that improve the application's performance. Customised hardware trades off flexibility for efficiency, enabling applications to run faster, reducing energy consumption, or both [Leiserson et al. 2020]. Hardware customisation could be key to improving application performance in edge devices.

FPGA (field programmable gate array) technology is a potential key enabler in that scenario. A FPGA is composed of building blocks, e.g., processor, memory and IO, connected by logic cells. One key advantage of FPGA is that the hardware configuration can be made dynamically and on demand by selecting which building blocks will be used. Attaching FPGAs into edge nodes could offer tailored hardware for ML applications at the edge. That hardware customisation in edge can improve the efficiency of ML implementations, accelerating "decision-making, hypothesis testing and even enable just-in-time interventions" [Fahim et al. 2021]. In this context, the open-source HLS4ML

framework [Fahim et al. 2021] translates trained ML models into FPGA implementations using high-level synthesis (HLS) tools.

In the context of IoT in Smart Cities, mobile devices like wearables, smartphones, or smart devices such as vehicles and drones require continuous connectivity while moving among different sites. In such a scenario, seamless mobility management strategies need to be developed, including seamless handover, service migration strategies, and dynamic resource allocation approaches to provide load balance.

Despite the lower computing power at the edge being mitigated by hardware customisation provided by FPGAs, providing that customisation in every edge node a mobile user connects is challenging. In this project, we propose a solution which mitigates that issue. Using the HLS4ML framework to translate ML models into HLS projects, which describes customised settings for ML implementation in FPGA devices, users can carry out such hardware configurations and apply them to different edge nodes. By carrying hardware settings, mobile users can use customised hardware to execute applications in a performance-enhanced environment. This work presents a discussion regarding that proposal. Section 2 presents two use cases that take advantage of that approach. Section 3 discusses the proposed scenario, and Section 4 presents the conclusions.

2. Use cases

This section presents two use cases that take advantage of both machine learning decision-making and edge computing resource provisioning as examples of the applicability of the proposed solution. Many applications in smart cities scenario fit that scenario. However, we focus on those with user mobility, high processing, and low latency requirements.

Case 1: Live Video Analytics for Drones. Identifying and taking action to avoid or minimise violent conflicts and crimes, especially in crowded spaces, is a security concern in cities. In a smart city, video surveillance systems can be improved by adopting unmanned aerial vehicles (UAVs), mainly characterised by drones. In such cases, drones with video cameras record a view from the top of a coverage area, feeding decision-making platforms and allowing managers to visualise and monitor target objects and people. However, due to the limited computing power of such vehicles, the data needs to be processed on remote servers. Due to the high volume of data and the real-time processing requirement, cloud servers may be impractical. In this scenario, data is sent to an edge infrastructure on the ground, which processes the video analytics. That processing is powered by machine learning algorithms. Due to drone mobility or any outage in their performance or availability, applications running on edge may need to be reallocated to different servers to keep the service uninterrupted. In this scenario, customised hardware and its customisation state migration could improve application performance.

Case 2: Intelligent transportation systems (ITS). Minimising traffic jams and improving driver and passenger safety are part of the urban mobility plan of every city. ITS join information technology, transportation and transit systems to provide better resource use for different transportation actors. Typically, these include cars, trains, bicycles, and road traffic lights, to cite a few. In that context, a vehicle equipped with video cameras, distance sensors and communication technologies could collect data from different sources and send it to be processed in edge servers near roads. Large-scale systems can use edge and cloud resources to process and manage ITS. Pattern recognition tasks could

be used in the perception of road signs, detection of pedestrians, vehicle detection in the surrounding area, and recognition of the actions of the driver and other actors, such as braking, steering, accelerating, lane changing, or crossing lanes [Yuan et al. 2022]. Similar to unmanned aerial vehicles, ground vehicles such as cars present a high mobility pattern and need remote computing resources to process robust applications. The proposed solution present in this work could improve application performance in edge devices by providing customised hardware for mobile users anywhere.

3. Proposed scenario

This work considers four scenarios in the performance evaluation of the proposed approach. Figure 1 illustrates those scenarios. The first one, named Scenario A in Figure 1 (a), ML applications run on generic hardware. Figure 1 (b) illustrates Scenario B in which every edge node that serves a mobile user along his/her path builds a hardware customisation to serve him/her. In that scenario, a hardware state is set in every user's connection. In Scenario C (Figure 1 (c)), the hardware customisation is set once. Then, that hardware state will be carried by the user and applied in his/her following edge servers along the path. Finally, Figure 1 (d) illustrates scenario D. Similarly to Scenario C, the customised hardware state is built once in Scenario D. However, that state is sent to the cloud before being applied to the following edge servers.

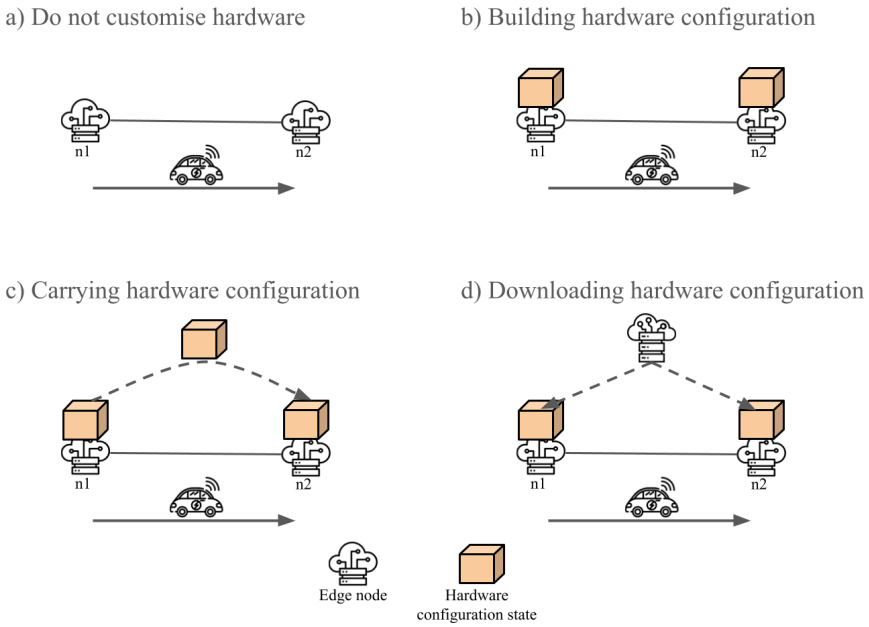

Figure 1. Hardware customisation scenarios for ML-based applications at the edge. a) baseline: it does not provide hardware customisation, i.e., use generic setup to run ML applications; b) Building hardware customisation in every user connection; c) Build hardware customisation once and then the user carries that state along his/her edge servers and; d) Build hardware customisation once and download it from the cloud in every edge server that the user is connected.

4. Final Remarks

In the context of user mobility support in edge infrastructures, different approaches have been proposed to improve the resource management and the applications' QoS, from fol-

low me cloud [Taleb et al. 2016] to Dynamic Network Slicing [Gonçalves et al. 2020]. However, none of them explores hardware optimisation. On the other hand, some solutions have been presenting improvements in the performance of machine learning applications based on hardware accelerators [Fahim et al. 2021, Ramchandani et al. 2023]. Nevertheless, no one has investigated that improvement in the mobile user context nor the follow-me cloud approach. This work in the initial phase proposes the use of hls4ml [Fahim et al. 2021] framework in FPGA-accelerated edge infrastructures to provide customised hardware for mobile users anywhere. Vivado HLS[1] and hls4ml can be used for FPGA performance evaluations and MobFogSim [Puliafito et al. 2020] for user mobility and resource management at edge computing infrastructures.

Acknowledgments

The authors thank Ioan Petri and Omer Rana from Cardiff University for their contributions to this work. This research is funded by FAPESP grant 19/26702-8. This work is part of the INCT of the Future Internet for Smart Cities (CNPq 465446/2014-0, CAPES 88887.136422/2017-00, and FAPESP 2014/50937-1). Diogo Gonçalves thanks CNPq grant 420907/2016-5, FAPESP #2015/24494-8, and CAPES finance Code 001.

References

Fahim, F., Hawks, B., Herwig, C., Hirschauer, J., Jindariani, S., Tran, N., Carloni, L. P., Di Guglielmo, G., Harris, P., Krupa, J., et al. (2021). hls4ml: An open-source codesign workflow to empower scientific low-power machine learning devices. *arXiv preprint arXiv:2103.05579*.

Gonçalves, D., Puliafito, C., Mingozzi, E., Rana, O., Bittencourt, L., and Madeira, E. (2020). Dynamic network slicing in fog computing for mobile users in mobfogsim. In *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, pages 237–246. IEEE.

Leiserson, C. E., Thompson, N. C., Emer, J. S., Kuszmaul, B. C., Lampson, B. W., Sanchez, D., and Schardl, T. B. (2020). There's plenty of room at the top: What will drive computer performance after moore's law? *Science*, 368(6495):eaam9744.

Puliafito, C., Goncalves, D. M., Lopes, M. M., Martins, L. L., Madeira, E., Mingozzi, E., Rana, O., and Bittencourt, L. (2020). Mobfogsim: Simulation of mobility and migration for fog computing. *Simulation Modelling Practice and Theory*, 101:102062.

Ramchandani, D., Asgari, B., and Kim, H. (2023). Spica: Exploring fpga optimizations to enable an efficient spmv implementation for computations at edge. In *2023 IEEE International Conference on Edge Computing and Communications (EDGE)*, pages 36–42. IEEE.

Taleb, T., Ksentini, A., and Frangoudis, P. A. (2016). Follow-me cloud: When cloud services follow mobile users. *IEEE Transactions on Cloud Computing*, 7(2):369–382.

Yuan, T., Da Rocha Neto, W., Rothenberg, C. E., Obraczka, K., Barakat, C., and Turletti, T. (2022). Machine learning for next-generation intelligent transportation systems: A survey. *Transactions on emerging telecommunications technologies*, 33(4):e4427.

[1]https://www.xilinx.com/products/design-tools/vivado.html