ML-Based Modeling and Virtualization of Reconfigurable Multi-Accelerator Systems

Juan Encinas

Centro de Electrónica Industrial, Universidad Politécnica de Madrid, Calle de José Gutiérrez Abascal 2, Madrid, Spain

Abstract

The work of this thesis focuses on providing reconfigurable multi-accelerator systems with the ability to self-adapt at run-time to the conditions and requirements of an IoT environment in a way that is transparent to the user. To this end, we have been working on an offline characterisation of the power consumption and performance of this type of systems through the development of a monitoring infrastructure and the production of predictive models based on machine learning techniques, obtaining very promising results. Currently the development is focused on converting this characterisation into an online modeling that allows, together with an already developed management infrastructure, to evaluate and validate this approach in a realistic test environment, and in the future we will work on the development of virtualization techniques for reconfigurable multi-accelerator systems that allow sharing the hardware among multiple tenants and applications, managing resources in an optimal and transparent way for the user and guaranteeing the performance, privacy and security of the system.

Keywords

Multi-Accelerator Systems, Reconfigurable Computing, Machine Learning, System Modeling, Virtualization ceuTechniques

1. Motivation and Objectives of the Thesis

The goal of this thesis is to develop design methodologies, support tools and decision making algorithms to provide reconfigurable multi-accelerator systems with the ability to adapt autonomously and at run-time to varying application conditions, environment and input data, in a transparent way to the user. Thanks to this self-adaptation capability, FPGA-based systems can be used as accelerators capable of handling computational requests from sensors or devices at the edge of the Internet of Things (IoT). This mechanism, known as computing offloading, allows processing to be brought closer to the points where the data is produced, offering higher processing performance, greater privacy, lower latency and lower power consumption than offloading to a remote cloud [1, 2]. Therefore, a scenario arises in which FPGA systems offer their processing power to the network, which is referred to as Acceleration-as-a-Service (AaaS) [3].

In order for FPGA-based systems to operate in this type of scheme, two problems have been identified: the decision making to optimally manage the available reconfigurable resources, as well as the virtualization of the FPGA's logical resources, thus isolating the application developer from the low-level details of the reconfigurable device used.

4th Workshop on Cyber-Physical Systems, September 19th, Sardegna, Italy

juan.encinas@upm.es (J. Encinas)
0000-0003-4727-7536 (J. Encinas)

1.1. Real-Time Modeling and Management of Reconfigurable Multi-Accelerator Systems

The inclusion of hardware accelerators in processing systems allows to improve their performance, both in terms of execution time and energy efficiency [4, 5]. In case multiple accelerators coexist to run one or multiple applications, we will speak of multi-accelerator systems.

In case all possible tasks to be accelerated in hardware are known in the design stages, they can be simulated and modeled analytically as part of the Design Space Exploration (DSE) process. However, in the computational offload scenarios detailed above, the hardware accelerators that will be required throughout the lifetime of the system cannot be known, nor the instants of time at which they will be demanded by the various network edge elements, as these will vary their behavior based on the data received or environmental conditions. For this reason, reconfigurable multi-accelerator systems working in this type of scenario must be able to adapt dynamically, ensuring that the system is always working at its optimum point, both from the perspective of energy consumption and throughput achieved.

In order to optimally manage the reconfigurable resources available, it will be necessary to model each of the system's accelerators beforehand. However, the modeling of the accelerators using analytical techniques is extremely complex, since each accelerator in execution will interfere with the others, due to the fact that there are shared elements, such as memories, controllers or communication buses within the chip. Therefore, each combination of accelerators must be characterized.

This is why this thesis proposes the development of models of multi-accelerator systems using machine learning algorithms. These models should be updated with the data produced by the execution of new combinations of accelerators or even new accelerator functionalities. On the other hand, a monitoring infrastructure will be designed to allow run-time performance and power consumption measurements in this type of systems, using such data to train the aforementioned models.

Based on the extracted models, machine learning based decision making algorithms will also be proposed, which will be able to select the optimal working point of the system at any given time for a given configuration. The metrics obtained will be fed back to the models, to provide them with the ability to be incrementally updated, so that the reconfigurable multi-accelerator systems can dynamically adapt to the changing conditions of the environment and possible unforeseen configurations.

1.2. Support for Reconfigurable Multi-Accelerator System Virtualization

FPGAs have gained a lot of importance in recent years in the world of cloud and edge computing due to their flexibility, high performance and low power consumption [1, 2]. Moreover, unlike other computing platforms such as CPUs and GPUs that feature a fixed architecture, FPGAs can adapt their architecture to the requirements of any algorithm due to their flexible hardware. FPGA hardware can be reconfigured to obtain both spatial and temporal parallelism on a large scale. As a result, FPGAs are high performance and energy efficient computing platforms, which is key for edge or cloud computing scenarios where available resources are limited.

Although FPGAs offer great benefits over CPUs and GPUs, these benefits require certain compromises in both design and usability. The application design flow for FPGAs requires the use of Hardware Description Languages (HDLs) and knowledge of the specific hardware at a low level, which restricts its use for most software application developers. Although High-Level Synthesis (HLS) tools are now available that allow FPGA applications to be developed using code with C-like syntax, it is still necessary to know certain hardware details to develop an optimized accelerator. In addition, the design process is specific to the hardware for which it is designed (obtaining a different binary depending on the FPGA model to be used) and the tools that vendors provide for design do not allow the use of system resources to be shared among multiple applications or users, which is essential for cloud and edge computing.

In this thesis we propose to design a virtualization infrastructure for FPGAs that allows both sharing resources among multiple applications and users, and designing applications without requiring specific knowledge of the FPGA hardware.

Specifically, we intend to explore and evaluate different existing virtualization techniques, both those used for software virtualization (the vast majority) and the approaches already proposed for hardware virtualization, and extend them in order to obtain an infrastructure with four key blocks:

1. Create an abstraction layer over the hardware to hide hardware-specific details from application designers and generate simple interfaces to access resources.
2. Manage the allocation of resources both spatially and temporally to make optimal use of resources.
3. Manage system resources in a way that is transparent to the user.
4. Ensure isolation between users and applications in terms of both performance and privacy, to guarantee data security and system resilience.

2. Thesis Progress

This section briefly describes the work accomplished to date, as well as the tasks planned for the remainder of the thesis.

2.1. Accomplished Works

At the present time, we have mainly focused on the real-time modeling and management of reconfigurable multi-accelerator systems.

We have design a monitoring infrastructure for acquiring power/performance traces in reconfigurable multi-accelerator systems. Those traces have been used for training offline ML-based models to predict power consumption and performance of such systems, obtaining promising results. And recently, we have developed a management infrastructure that will be used to integrate and validate the rest of this part of the thesis.

2.1.1. Monitoring Infrastructure

A non-intrusive monitoring infrastructure has been design to acquire synchronized power consumption and performance traces in reconfigurable multi-accelerator systems (see Figure 1). To do so, hardware components (block diagram of the infrastructure in HDL), software components (driver for managing the infrastructure's hardware from a Linux-based OS and libraries for controlling the infrastructure on Linux-/baremetal-systems), as well as a script-based visualization tool for visualizing the generated traces have been designed.

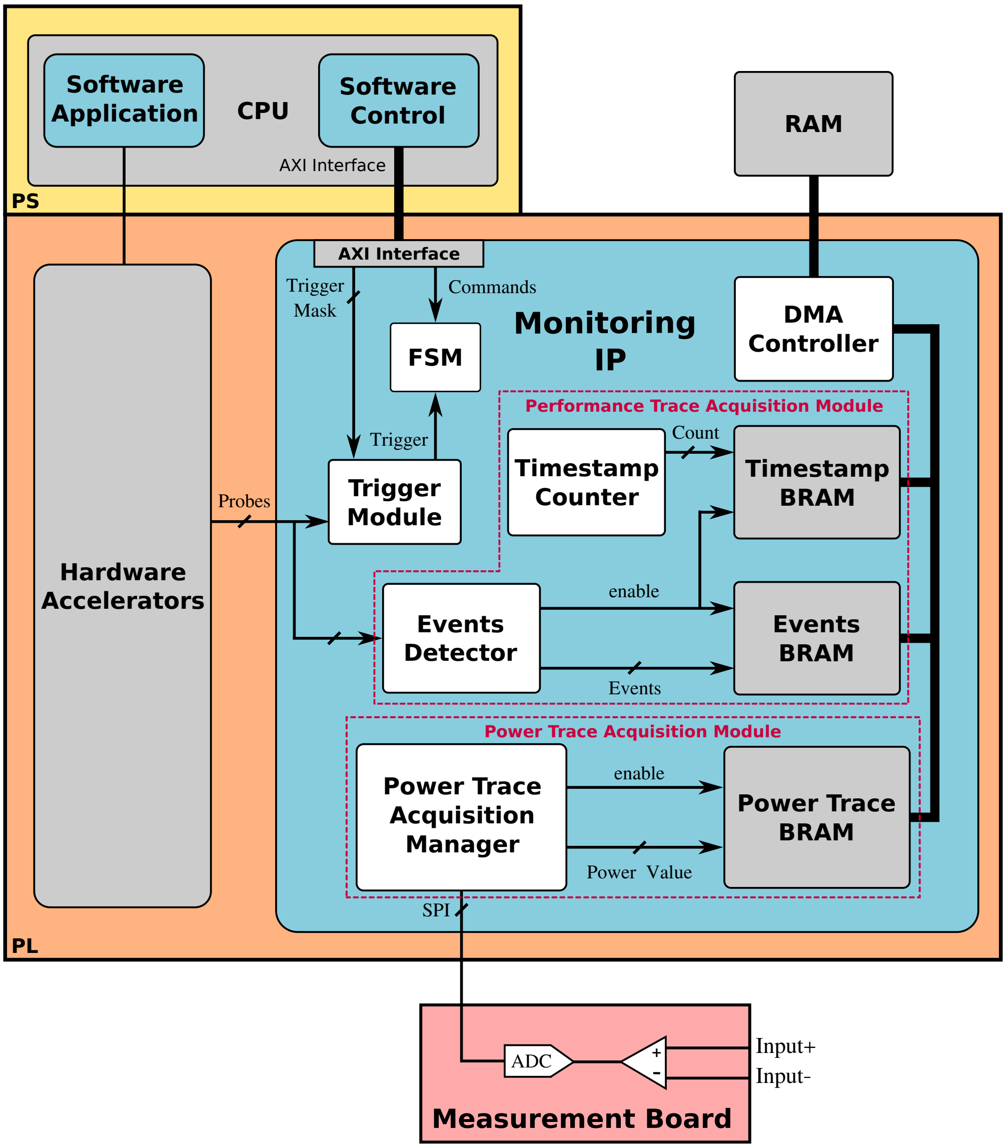

Figure 1: General Overview of the Monitoring Infrastructure.

Figure 2 show an example of the traces generate with the infrastructure, where an application with 2 hardware accelerators that exhibit different functionality is depicted (signals 0/1 and 2/3 are the start/ready pairs for accelerators A and B, respectively, and power consumption is shown at the top).

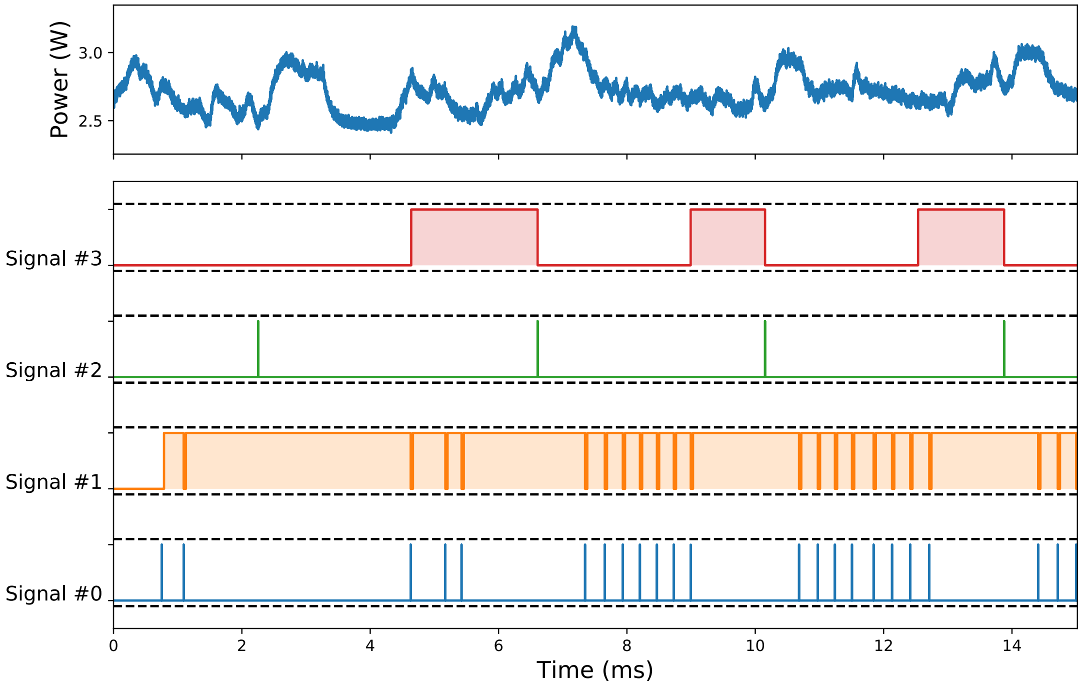

Figure 2: Example of Power Consumption and Performance Traces.

2.1.2. ML-Based Modeling

In order to properly model the power consumption and performance of reconfigurable multi-accelerator systems we have obtained, with the monitoring infrastructure, power and performance traces of multiple combinations of hardware-accelerated kernels from the MachSuite [6] benchmark suite, a well-known benchmark suite for HLS-oriented accelerator evaluation. The obtained traces have been used to train ML-based models at predicting power consumption and performance. Those models have been subsequently evaluated, with very good results. As a example Figure 3 and Figure 4 show the graphical evaluation of a particular model when predicting power consumption and performance, respectively.

In both figures, the predicted value of each observation is plotted against its actual value (the dotted diagonal line represents ideal points where the predicted value equals the measured value). It can be observed that in both cases most of the observations fall really close to the dotted line indicating that the model has a good prediction performance. For a more in-depth analysis, refer to our paper on the subject [7].

2.1.3. Management Infrastructure

A management infrastructure has been design, capable of attending to all the incoming acceleration requests of a particular workload, deciding when to execute them in the FPGA fabric following a specific scheduling policy (see diagram in Figure 5).

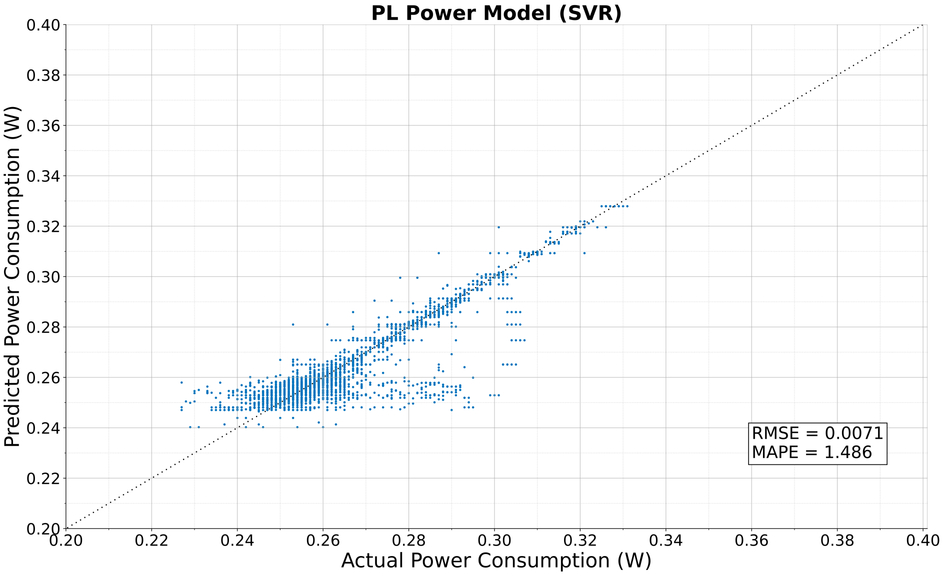

Figure 3: PL Power Model Evaluation.

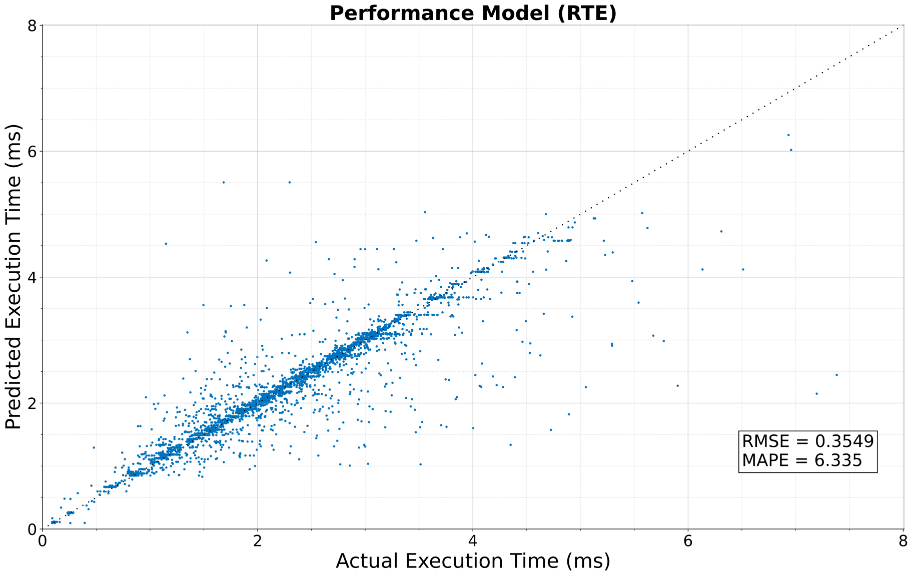

Figure 4: Performance Model Evaluation.

For the hardware acceleration we have extended the ARTICo^3 framework [8], an academic framework for high-performance reconfigurable multi-accelerator system implementation. We have also integrated the monitoring infrastructure described above, enabling the monitoring of every part of the process.

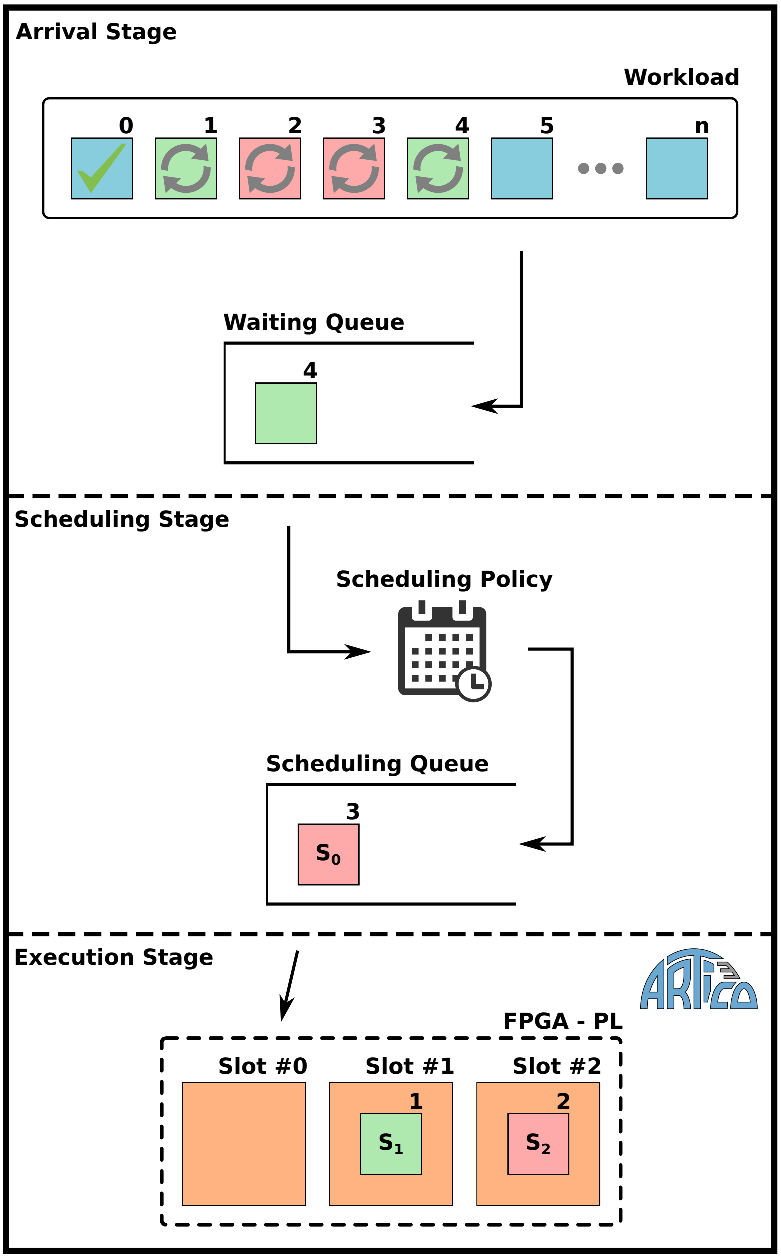

Figure 5: Management Infrastructure.

2.2. Future Work

We are currently working on doing an online model training and update rather than an offline characterization as a first step to achieve full run-time self-adaptation. And we are also thinking about the idea of integrating complex scheduling policies within the management infrastructure, such as reinforcement learning decision making and other alternative approaches based on the online data-driven models, to perform an intelligent decision making on the task scheduling and resource management of the system.

This would conclude the first pillar of the thesis and we would then focus on the FPGA virtualization part described in Section 1.

3. Acknowledgements

This thesis is receiving funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 872570.

References

[1] C. Xu, S. Jiang, G. Luo, G. Sun, N. An, G. Huang, X. Liu, The case for fpga-based edge computing, IEEE Transactions on Mobile Computing (2020).

[2] L. Cerina, S. Notargiacomo, M. G. Paccanit, M. D. Santambrogio, A fog-computing architecture for preventive healthcare and assisted living in smart ambients, in: 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), IEEE, 2017, pp. 1–6.

[3] C. Bobda, J. M. Mbongue, P. Chow, M. Ewais, N. Tarafdar, J. C. Vega, K. Eguro, D. Koch, S. Handagala, M. Leeser, M. Herbordt, H. Shahzad, P. Hofste, B. Ringlein, J. Szefer, A. Sanaullah, R. Tessier, The future of fpga acceleration in datacenters and the cloud, ACM Trans. Reconfigurable Technol. Syst. 15 (2022). URL: https://doi.org/10.1145/3506713. doi:10.1145/3506713.

[4] M. Qasaimeh, K. Denolf, J. Lo, K. Vissers, J. Zambreno, P. H. Jones, Comparing energy efficiency of cpu, gpu and fpga implementations for vision kernels, in: 2019 IEEE International Conference on Embedded Software and Systems (ICESS), 2019, pp. 1–8. doi:10.1109/ICESS.2019.8782524.

[5] S. Asano, T. Maruyama, Y. Yamaguchi, Performance comparison of fpga, gpu and cpu in image processing, in: 2009 International Conference on Field Programmable Logic and Applications, 2009, pp. 126–131. doi:10.1109/FPL.2009.5272532.

[6] B. Reagen, R. Adolf, Y. S. Shao, G. Wei, D. Brooks, MachSuite: Benchmarks for accelerator design and customized architectures, in: 2014 IEEE International Symposium on Workload Characterization (IISWC), 2014, pp. 110–119. doi:10.1109/IISWC.2014.6983050.

[7] J. Encinas, A. Rodríguez, A. Otero, E. De La Torre, Run-time monitoring and ml-based modeling in reconfigurable multi-accelerator systems, in: 2021 XXXVI Conference on Design of Circuits and Integrated Systems (DCIS), 2021, pp. 1–7. doi:10.1109/DCIS53048.2021.9666187.

[8] A. Rodríguez, J. Valverde, J. Portilla, A. Otero, T. Riesgo, E. de la Torre, FPGA-Based High-Performance Embedded Systems for Adaptive Edge Computing in Cyber-Physical Systems: The ARTICo^3 Framework, Sensors 18 (2018). doi:10.3390/s18061877.