Hybrid Performance Prediction Models for Fully-Connected Neural Networks on MPSoC

Authors: Quentin DARIOL[12] (quentin.dariol@etu.univ-nantes.fr), Sébastien LE NOURS[1], Sébastien PILLEMENT[1], Ralf STEMMER[2], Domenik HELMS[2] and Kim GRÜTTNER[2]

[1] : Nantes Université, IETR UMR CNRS 6164 | [2] : German Aerospace Center (DLR)

Motivation

Context

- Fast growth of the need for **Artificial Neural Networks (ANNs)** on **embedded** devices with **limited resources**,
- Required exploration flow to identify **optimized implementations**.

Challenge

- On **multi-core platforms, concurrent accesses to shared resources** surge the latency and are hard to model,
- Other approaches fail to properly evaluate the **impact of communications** on timing for a large scale of architectures.

Objective

- Propose a **hybrid modeling flow** to enable fast and accurate **performance prediction** for **ANNs on Multi-Processor Systems on Chip (MPSoCs)**.

Contribution

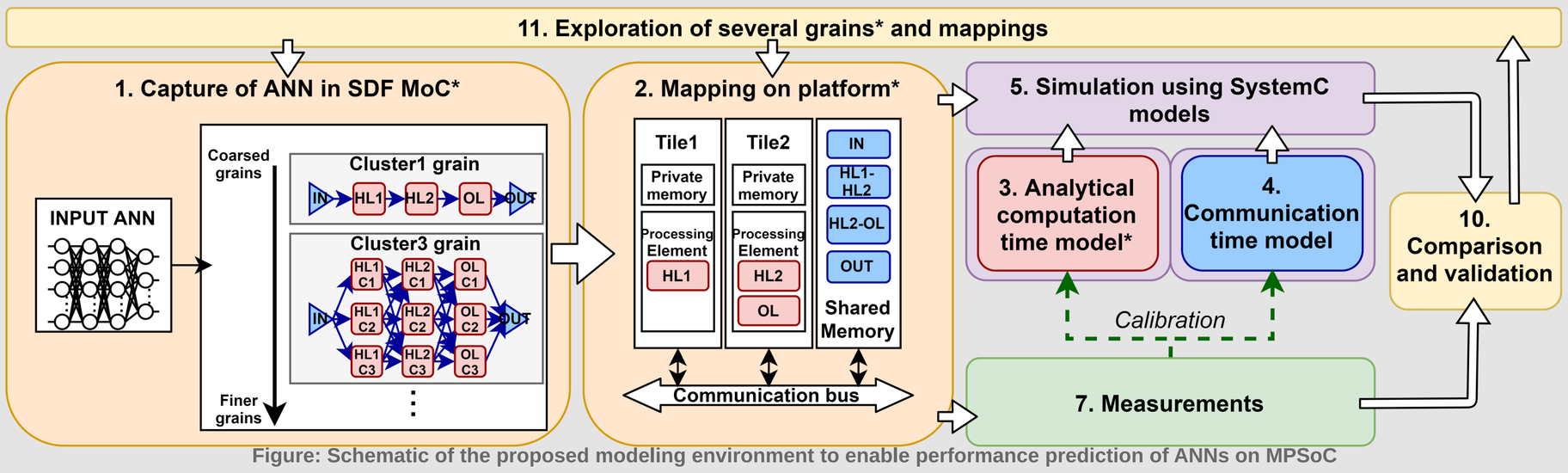

Figure: Schematic of the proposed modeling environment to enable performance prediction of ANNs on MPSoC

Modeling environment

- Extension of the environment proposed in previous work to ANNs,
- Proposition of an **analytical computation time model calibrated through measurement**, used in a **SystemC simulation,**
- **Exploration** of several partitionings and mappings of ANNs **under timing constraints** using the modeling flow

Case study

- **Fully-connected ANNs** (also called multi-layer perceptrons) trained using the MNIST and the GTSRB data-sets,
- ANNs are described using the **deep learning framework LibFANN** and modeled using **Synchronous DataFlow (SDF)**,
- Targeted architectures consist of a **shared memory and several tiles** composed of 1 processing element with local memory.

Validation

- Validation of the modeling flow by **comparing timing predictions with real measurements** for several partitionings and mappings of **three fully-connected ANNs** on multi-core architectures implemented on **two different FPGAs**,
- The observed **accuracy** of the modeling flow is **99,5%** on average on 21 different scenarios, with **highest error of 2,28%.**

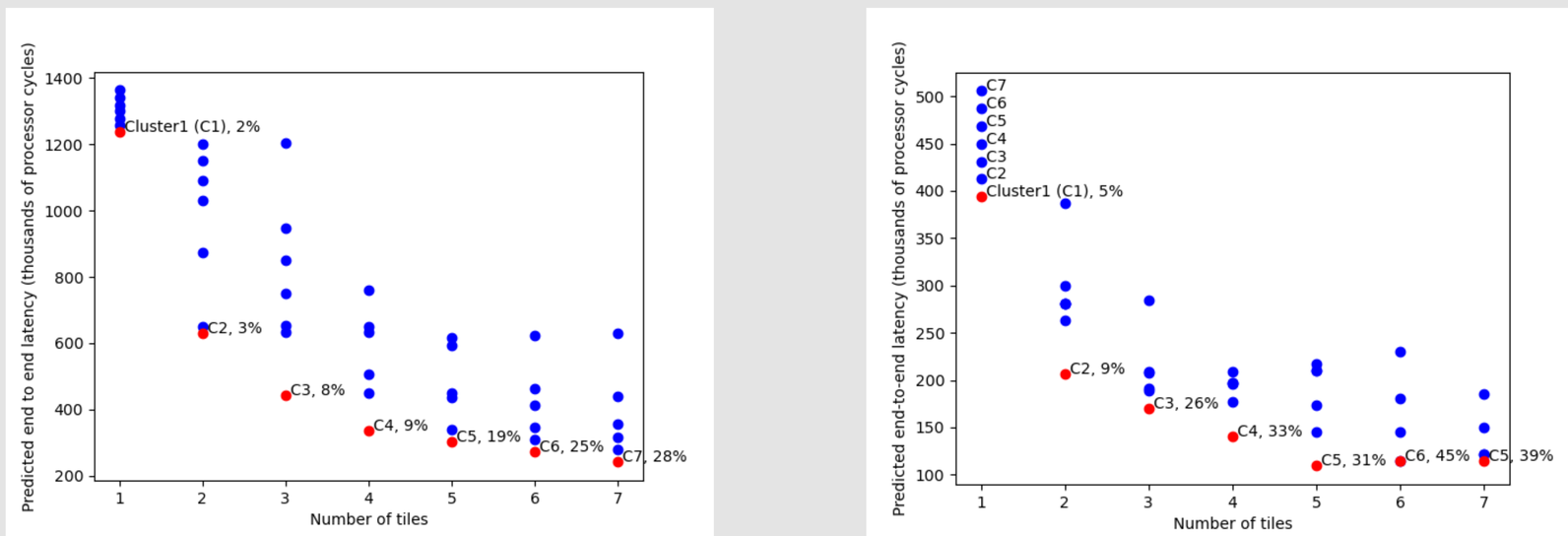

Figure: Exploration of several partitionings and mappings for two different ANNs and identification of optimized solution for each number of tiles (highlighted in red)

Prospects

- Proposition and integration of a **power model** in the modeling environment to enable **exploration under timing and energy constraints,**
- Extension of the flow to **other classes of ANNs** such as **Convolution Neural Networks (CNNs).**

This work has been funded by the WISE consortium, France (pSSim4AI project) and by the Federal Ministry of Education and Research (BMBF, Germany) in the project Scale4Edge (16ME0465).

DLR Deutsches Zentrum für Luft- und Raumfahrt

IETR INSTITUT D'ÉLECTRONIQUE ET DE TÉLÉCOMMUNICATIONS DE RENNES

Hybrid Performance Prediction Models for Fully-Connected Neural Networks on MPSoC

Quentin DARIOL[1,2], Sebastien LE NOURS[1], Sebastien PILLEMENT[1],
Ralf STEMMER[2], Domenik HELMS[2], and Kim GRÜTTNER[2]

[1]Nantes University, IETR UMR CNRS 6164, France, Email: quentin.dariol@univ-nantes.fr
[2]German Aerospace Center (DLR), Germany

***Abstract*—Predicting the performance of Artificial Neural Networks (ANNs) on embedded multi-core platforms is tedious. Concurrent accesses to shared resources are hard to model due to congestion effects on the shared communication medium, which affect the performance of the application. In this paper we present a hybrid modeling environment to enable fast yet accurate timing prediction for fully-connected ANNs deployed on multi-core platforms. The modeling flow is based on the integration of an analytical computation time model with a communication time model which are both calibrated through measurement inside a system level simulation using SystemC. The proposed flow enables the prediction of the end-to-end latency for different mappings of several fully-connected ANNs with an average of more than $99\,\%$ accuracy.**

***Keywords* — Model of Performance, Multi Processor, SystemC simulation, Artificial Neural Networks**

I. Introduction

The demand for smart multi-core embedded systems used to execute Artificial Neural Networks (ANNs) has increased exponentially in the last few years. Deploying ANNs on such devices is tough as ANNs are computation-intensive application with high memory needs whereas embedded platforms lack of computational resources and memory, and often bear strong timing and energy constraints. To ease the development effort and allow the optimization of the implemented solutions, early performance evaluation for ANNs on MPSoC is needed.

Several approaches have been carried out to tackle the challenge of performance evaluation for ANNs on embedded platforms. Some approaches focus on evaluation through implementation and testing [1] [2]. This evaluation technique doesn't enable a comprehensive exploration of the design space due to the long development time needed to systematically deploy the ANN. Other works focus on the building of analytical models [3] [4], however such models lack of accuracy when targeting a large scale of architectures composed of multiple processing elements. On these architectures concurrent accesses on shared elements can cause congestions, which surge execution time and are hard to model.

To tackle this challenge we present in this paper an innovative modeling environment used for fast-yet-accurate performance prediction for fully-connected ANNs, also called Multi-Layer Perceptrons (MLPs), deployed on MPSoC. This modeling environment combines simulation, analytical models and partial characterization through measurements. We demonstrate the effectiveness of our flow by predicting the performance of several partitionings and mappings of two MLPs on architectures containing up to 7 cores with an average accuracy of more than $99\,\%$.

II. Modeling and simulation environment

A schematic of the proposed environment is given in Fig. 1. This figure highlights the steps to build performance models used to predict execution time of MLPs deployed on multi-core platforms. The first activity captured in orange focuses on the description of the MLP in the Synchronous Data Flow (SDF) Model of Computation (MoC) (1.) and its mapping on the targeted platform (2.). SDF offers a strict separation of computation and communications (read, write) of actors. Computation and communication separation eases the performance prediction process by allowing building separated computation time and communication time models. To apply the SDF MoC to fully-connected ANNs, we define a cluster as a given set of neurons from the same layer, which execution is modeled as an actor. The number of actors (i.e. the number of clusters) by layer is established based on the desired granularity. Two examples of SDF graph of different complexity are given in Step 1. of Fig. 1. The next step aims at mapping the ANN on the targeted architecture. The considered architecture is composed of a set of tiles. A tile is one processing element with its private memory. Actors of the SDF graph are mapped on the tiles and communication channels are mapped on the shared memory. An example of mapping is depicted in Step 2. of Fig. 1.

The next activity highlighted in purple focuses on the building of a simulable model (5.) using the developed computation and communication time models (3. and 4.). The computation time model is an hybrid analytical model calibrated by measurements. It can predict accurately performance for any considered level of granularity used to describe ANN applications. The communication time model is a message level model that was introduced in previous work [5]. Both the computation time model and the communication time model are integrated in a high level simulation of the architecture executing the application. This allows to perform a timing analysis of the ANN executed on the targeted platform.

The third activity highlighted in green focuses on the measurement of delays on a real platform (6.) in order to

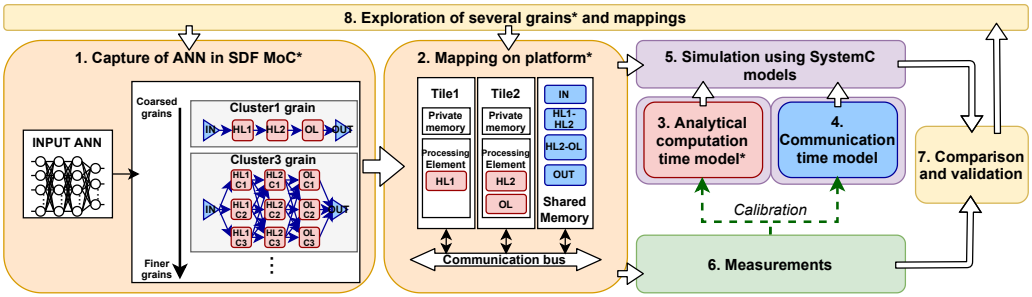

Fig. 1: Schematic of the proposed environment. This flow aims at enabling performance prediction for fully-connected ANNs on multi-core platforms. The processes marked with the * symbol are novel features to our flow.

calibrate the timing models and compare the simulation results with real execution times. The final activity highlighted in yellow focus on the validation of the proposed models by comparing the predictions to the measured execution times (7.) and the exploration of several grains and mappings to optimize latency using the proposed models (8.).

III. Experimental Setup and Results

To characterize our models of performance and to validate their predictions we implemented a hardware platform that followed the hypothesis of the proposed model of architecture. We implemented this platform on both a Xilinx Ultrascale MPSoC+ and a Xilinx Zynq7000 FPGAs. A tile is composed of a MicroBlaze processor and a local memory implemented as BRAM. The platform is composed of 7 tiles and a shared memory connected via an AXI shared interconnect. It implements a time measurement infrastructure which allows measuring the delays needed for the whole SDF graph execution or actor execution. The computation and communication time models have been calibrated with elementary delays measured using this infrastructure. In this work, the ANNs are implemented using the lightweight open source library LibFANN [6] which enables the training and execution of MLPs in C programming language. To verify and validate our modeling approach, we considered two MLPs with different topologies. Both applications were developed and trained to perform digit recognition using the MNIST data set.

In order to test the accuracy of the proposed models for several levels of complexity of SDF graphs, a set of SDF graphs with different grains were used for both applications. Several mappings were also considered for each of these SDF graphs, which uses up to 7 tiles and rely both on parallel and pipeline execution. The predicted end-to-end latencies by the proposed simulable model are compared with the end-to-end latencies measured on the real platform for several scenarios. We also compared the predictions of the computation time model used alone against the real durations. The predictions of the computation time model only are too optimistic and bear $10.5\,\%$ error on average for the presented scenarios. In order to predict the execution time of ANNs on multi-core platforms, a computation time model alone is not sufficient and the modeling of communications is necessary. The system level simulable model averages more than $99\,\%$ accuracy for all the considered scenarios. The highest prediction error is $2.89\,\%$ for a complex 7-tiles scenario that benefits both from parallel and pipeline execution. In this scenario the average communication time of tiles counts for $89\,\%$ of the execution time. For all the other scenarios, the error of the simulable model is less than $1\,\%$. This validates the proposed model for the prediction of execution time of MLPs on multi-core platforms. Once validated, the modeling approach can be used to explore other partitionings and mappings and find optimized solutions.

IV. Conclusion

Our hybrid modeling environment enables accurate performance prediction for MLPs on MPSoC architectures containing up to 7 tiles. It achieves overall $99\,\%$ accuracy for estimating the end-to-end latency for two MLPs trained using the MNIST data-set. In future work, we will extend the presented modeling environment to enable the exploration of the design space for several classes of ANNs under both timing and energy constraints. We will propose a power and energy modeling flow that will be validated by comparing with real consumption measured on the execution platform, as well as evaluating the effect of power management methods on the execution time and energy consumption of ANNs.

References

[1] I. Galanis et al., "Inference and Energy Efficient Design of Deep Neural Networks for Embedded Devices," 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2020, pp. 36-41.

[2] F. Tsimpourlas et al., "A Design Space Exploration Framework for Convolutional Neural Networks Implemented on Edge Devices", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 2212-2221, Nov. 2018.

[3] A. Parashar et al., "Timeloop: A Systematic Approach to DNN Accelerator Evaluation," 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2019, pp. 304-315.

[4] Salita Sombatsiri et al., A design space exploration method of SoC architecture for CNN-based AI platform. SASIMI Proceedings, 2019.

[5] Ralf Stemmer et al. A measurement-based message-level timing prediction approach for data-dependent SDFGs on tile-based heterogeneous MPSoCs. Applied Sciences, 11(14), 2021.

[6] Steffen Nissen. Implementation of a fast artificial neural network library (fann), https://github.com/libfann/fann, 12 2003