Experiencing Self-Aware MPSoC Run-Time Optimization with Autonomous Bots

Florian Maurer, Thomas Wild, and Andreas Herkersdorf
Chair of Integrated Systems, Technical University of Munich, Germany

ABSTRACT

The effects of changes to the software or hardware in the field of autonomous and self-adaptive robotics can usually be perceived visually and haptically in three-dimensional space and can literally be experienced. Of course, these software and hardware changes also impact conventional metrics of embedded computing systems, such as performance in terms of instructions per second, memory access latencies, and power dissipation. Graphs over time are needed to visualize changes in these conventional metrics. But especially the aspect of experiencing design changes interactively and visually makes robotics attractive for students.

Building on the Duckietown robotics platform [1], we are working on use-cases that aim to visualize our research on reinforcement learning-based (RL-based) Learning Classifier Systems (LCTs) conventionally as well as visually and interactively. For these use cases, there are also graphs of memory access times and bus conflicts, but extended by the experienceable aspect of e.g., crashes of autonomously driving robots, and adhered or not adhered driving corridors.

A. Related Work

1) LCTs: So far, we have explored self-optimization of multiprocessor system-on-chips (MPSoCs) by extending them with our LCTs. Self-optimization is achieved by LCTs adapting run-time parameters (e.g., per core voltage and frequency, task mapping) of the MPSoCs in the microsecond range to changing conditions in order to meet the requirements of the embedded applications being executed. As an RL unit, LCTs need to be informed about the impact of their executed actions to improve their decision-making. The impact is derived from metrics of interest for the platform under control and provided as a numerical reward. In our MPSoC setups, implemented on Xilinx 7-series FPGAs, we calculate the reward based on the change in performance (IPS) and the current energy consumption [2].

2) Duckietown: Duckietown is a platform used worldwide for teaching and research in the context of autonomous driving. It was designed with this wide range of use-cases in mind, providing easy-to-use infrastructure for undergraduate students and a complex software architecture allowing researchers to conduct in-depth analyses. The bots' feature a monocular camera for sensing, two DC motors for moving and Nvidia Jetson Nano boards for computation.

We would like to thank for the financial support from the DFG under Grant HE4584/7-2.

B. Use-cases

By enhancing Duckietown bots by LCTs, we pursue two goals. First, we want to demonstrate the potential of MPSoC run-time optimization in a visual and haptic manner. Second, it allows us to showcase how applicable LCTs are to other domains. Actually, we started due to the easier implementation with setups belonging to the second case. Our initial setups resemble the adaptation of run-time parameters in order to fulfill an application's requirements. As with MPSoCs, energy consumption is an essential aspect of autonomous driving in order to maintain sufficient range. Additionally, passengers want to arrive at their destination on time, but the autonomous vehicle should also comply with traffic regulations. In our initial approach (a), we let an LCT control the frame rate of a bot's camera to reduce the amount of energy consumed by the image processing. In approach (b), an LCT decides on its bot's speed to arrive 'on time'. Both approaches use a software implementation of the LCTs, as their decisions are required less frequently compared to the MPSoC use-case.

From the approach description it is intuitive to reward approach (a) for low frame rates and approach (b) for high speed. However, driving according to traffic regulations has the highest priority, so we have to penalize losing track and crossing lines additionally.

C. Challenges and Future Work

We expect higher or lower speeds and frame rates depending on the driving context (straight vs. turn and the bot is centered to the lane vs. close to the road marking). Unfortunately, our initial experiments did not result in the expected outcome as the bots tend to crash relatively often already without our LCT extension. Hence, students improved the lane-following demo of Duckietown by making it more robust against light reflections and tuning the PI controller in its control loop.

As mentioned previously, our primary goal is to let students experience our research on run-time optimization of MPSoCs. Therefore, we port the image processing code to make use of the available GPU, and we are working on an FPGA extension that contains dedicated accelerators. Once this is accomplished, students can observe how different LCT configurations impact the mapping of tasks to CPU, GPU, and HW accelerator and hence change the bots' driving performance.

REFERENCES

[1] L. Paull and et al., "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1497–1504.

[2] F. Maurer and et al., "Machine learning in run-time control of multicore processor systems," *it - Information Technology*, vol. 65, no. 4-5, pp. 164–176, 2023. [Online]. Available: https://doi.org/10.1515/itit-2023-0056