Predicting ADAS algorithms performances on K1 architecture

Romain SAUSSARD[1,2] – Boubker BOUZID[1] – Roger REYNAUD[2] – Marius VASILIU[2]
[1]Renault, [2]Université Paris Sud

Introduction

- Computer Vision Algorithms are widely used in automotive field for the ADAS.
- A lot of computing architectures can be used to embed those algorithms : ARM, DSP, GPU and heterogeneous one like the K1. It's not easy to choose the best algorithm – architecture association.
- Existing models for performance prediction are only applicable on one architecture, not on heterogeneous systems. For example the one in [1] can be used only for CUDA.
- We propose a method to predict performance on multiple, heterogeneous architectures in order to help choosing the best algorithm – architecture association.
- We illustrate our approach with a lane detection algorithm embedded on different architectures.

Classes of Instructions

- An algorithm is a set of instructions, and each instruction can be classified.
- An architecture, a, has different throughput ($p_{c,a}$ in instructions per cycle) for each instruction class, c.
- C: set of computing instructions, $c \in C$.
- M: memory instructions (*Load & Store*).
- Ia : arithmetic intensity [2], number of operations for each memory instruction. This can be used to estimate the bottleneck [3].

$$Ia = \frac{N_C}{N_M} \qquad t_{max,a} = \sum_{i \in C} \frac{N_c}{p_{c,a}} \qquad t_{min,a} = \max_{\{i \in C\}} \left(\frac{N_c}{p_{c,a}}\right)$$

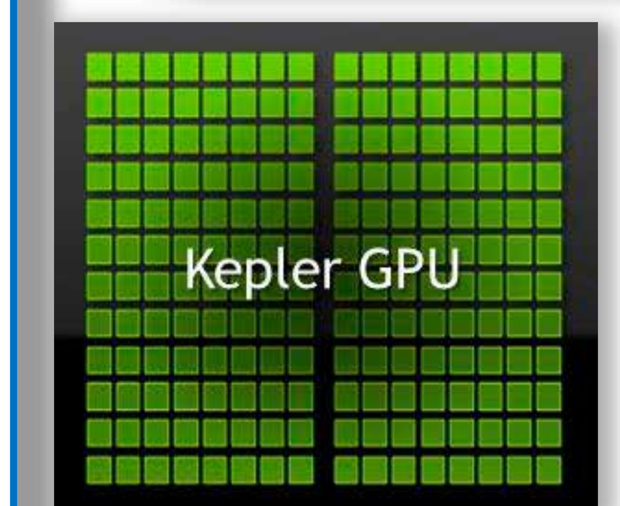
Throughput : $p_{c,a}$

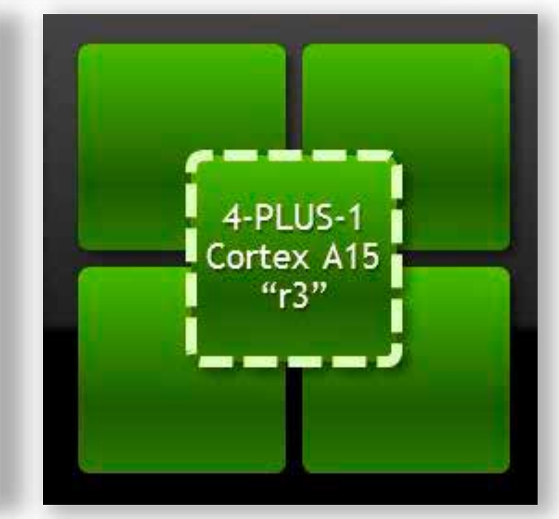
Instruction	ARM A15	CUDA/GPU K1
Simple Int	2	160
Mult. Int	1	32
Float	1	192
Specific	*	32
Branch	1	32
Address	1	160
NEON Load & Store	0.5	/

* Multiple instructions

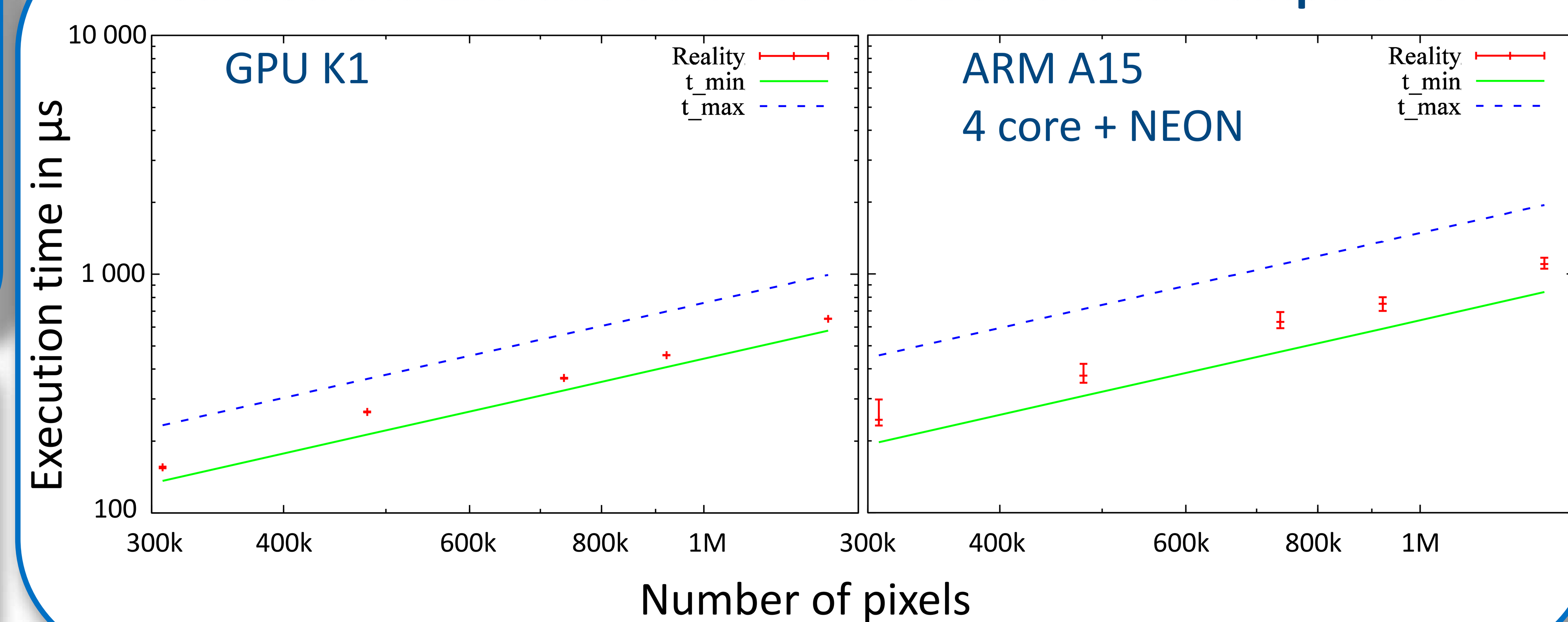
Predicted Results

Algorithm		ARM A15 1.5 GHz 1 core	ARM A15 1.5 GHz 4 cores + NEON	CUDA/ GPU K1 600 MHz
Gradient $Ia = 4$	t_{max}	19 Mp/s	380 Mp/s	490 Mp/s
	t_{min}	57 Mp/s	1030 Mp/s	900 Mp/s
	Reality	32 Mp/s	450 Mp/s	660 Mp/s
	Precision	±50%	±46%	±30%
Bottom Hat $Ia = 5$	t_{max}	18 Mp/s	670 Mp/s	1320 Mp/s
	t_{min}	32 Mp/s	1560 Mp/s	2260 Mp/s
	Reality	30 Mp/s	1250 Mp/s	2000 Mp/s
	Precision	±28%	±40%	±26%





Predicted and Execution time for the Bottom Hat computation



Lane detection algorithm

- Top left : Input image
- Top right : Gradient of the image
- Bottom left : Bottom Hat with a 1x5 structuring element

Conclusion and future work

- Our model is able to predict an execution time interval for heterogeneous architectures if the Ia is high enough.
- The model needs to be improved by taking into account memory delay for algorithms with small Ia.
- Apply our model for more complex algorithms, like parallel reduction.

[1] HONG, Sunpyo et KIM, Hyesoon. An analytical model for a GPU architecture with memory-level and thread-level parallelism awareness. In : *ACM SIGARCH Computer Architecture News*. ACM, 2009. p. 152-163.
[2] M. Harris. Mapping computational concepts to GPUs. In *ACM SIGGRAPH 2005 Courses*, page 50. ACM, 2005.
[3] S. Williams, A. Waterman, and D.Patterson. Roofline : an insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4) :65–76, 2009.