A Model-driven development framework for highly Parallel and EneRgy-Efficient computation supporting multi-criteria optimisation

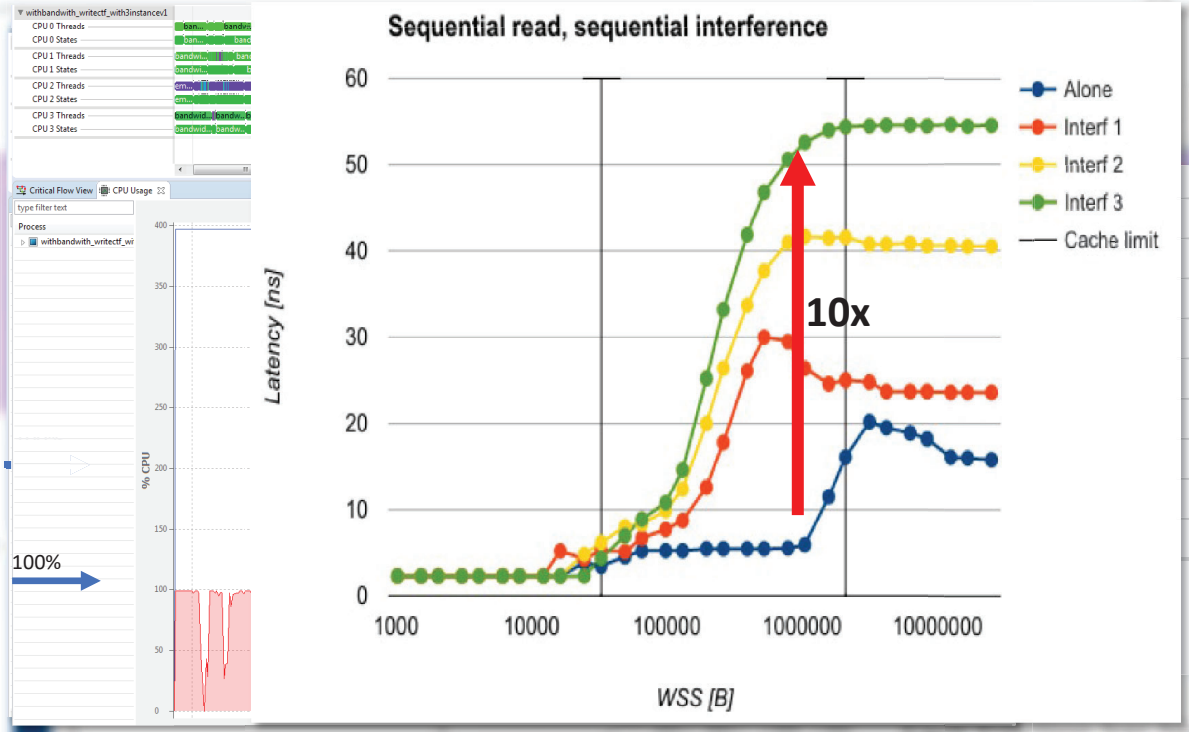
Model-Driven Engineering Use Cases: Automotive

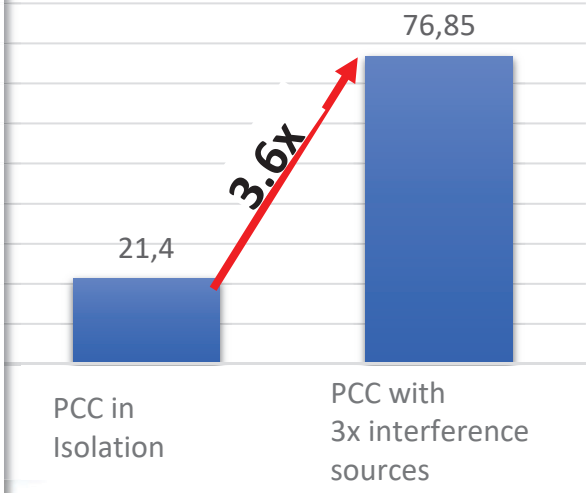
AMPERE Final Event Webinar

Michael Pressler — Robert Bosch GmbH

27 June 2023

2020 research and innovation programme under grant agreement No 871669





Source: Roberto Cavicchioli, Nicola Capodieci, Marko Bertogna, Memory interference characterization between CPU cores and integrated GPUs in mixed-criticality platforms. ETFA 2017

The Context

Hundreds of ECUs regulate everything, e.g., battery charge, fuel supply, climate control,…

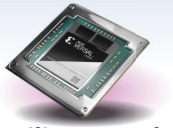
Higher level (ADAS) functions leverage data from lower-level functions

Vehicle/Domain computers integrate functions developed by different supplierers on the same System-on-Chip. Funtionality that **must operate correctly** in response to its inputs from both **functional** and **non-functional perspectives**

*Model-Driven Engineering based on **Domain Specific Modeling Languages** (DSML)*

***Parallel Programming Models** (PPM) for shared-memory and accelerator devices*

Massively parallel systems that operate **as fast as possible**



Xilinx Versal (FPGA w. DFX)

NXP S32G (Multi/Many-Core)

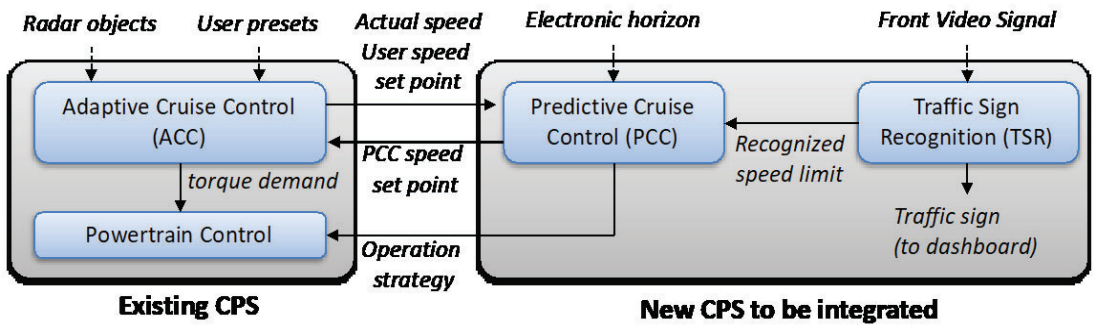
NVIDIA Jetson (GPU)

Complex processor architectures with different ISA, SDK, …

Constant evolution

Bosch: Intelligent Predictive Cruise Control

- **Three different application classes with different execution semantics**
  - ACC (Adaptive Cruise Control) & Powertrain Control
  - PCC (Predictive Cruise Control)
  - TSR (Traffic Sign Recognition)
- **Gluing it all together: Publish-Subscribe Middleware**
  - Autosar Adaptive
  - ROS2



Powertrain Control

- **Autosar Classic Semantics**
  - Periodic tasks, sequentially executing
  - Runnables, communicating on a fine grain level via variables in shared memory (a.k.a. labels)
  - WATERS Challenges 2016 & 2017
  - Executed on CPU
  - ASIL B

I	II	III	IV	V	VI
<10	10-50	51-100	100-500	501-1000	>1000

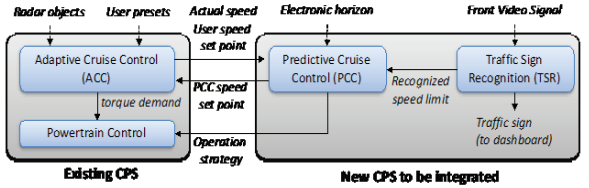
Period	1 ms	2 ms	5 ms	10 ms	20 ms	50 ms	100 ms	200 ms	1000 ms	sync
1 ms				I	I		I			I
2 ms				I	I		I			
5 ms		I	IV	IV	II	II	I			
10 ms	II	II	II	VI	IV	II	IV	II	III	IV
20 ms	I	I	I	IV	VI	II	IV	I	II	IV
50 ms			II	II	II	III	I			
100 ms		I	I	V	IV	II	VI	II	III	IV
200 ms				I	I		I	I	I	
1000 ms				III	II		III	I	IV	I
Angle-sync	I	I	I	IV	IV	I	III	I	I	V

Inter task communication

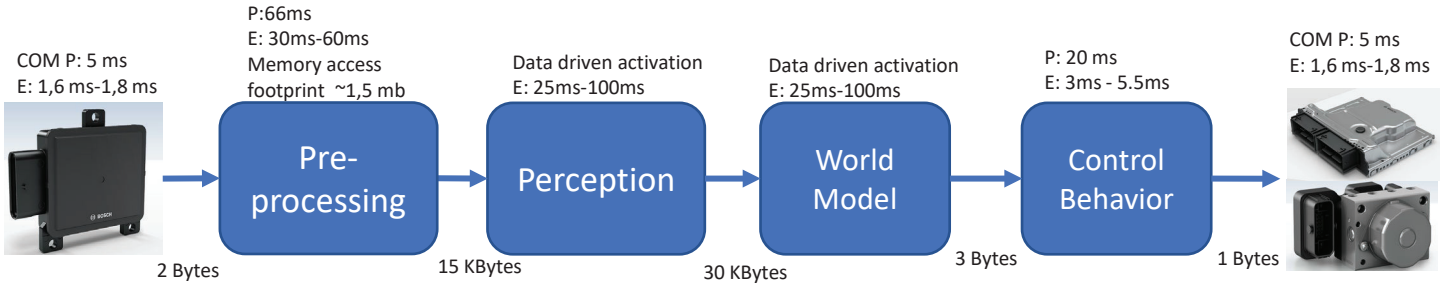
Amalthea Element	#
Tasks	~21
Runnables	~1250
Labels	~10000

Adaptive Cruise Control (ACC)

- **Autosar Classic Semantics**
  - Periodic tasks with data driven activation (sampling + pipelining)
  - Runnables, communicating on a fine grain level via variables in shared memory (a.k.a. labels)
  - Potential offloading of some functions to FPGA and GPU

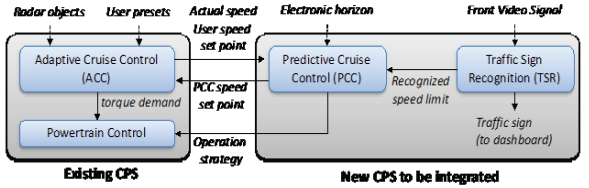


Amalthea Element	#
Tasks	~5
Runnables	~100 (w/o base SW)
Labels	~500

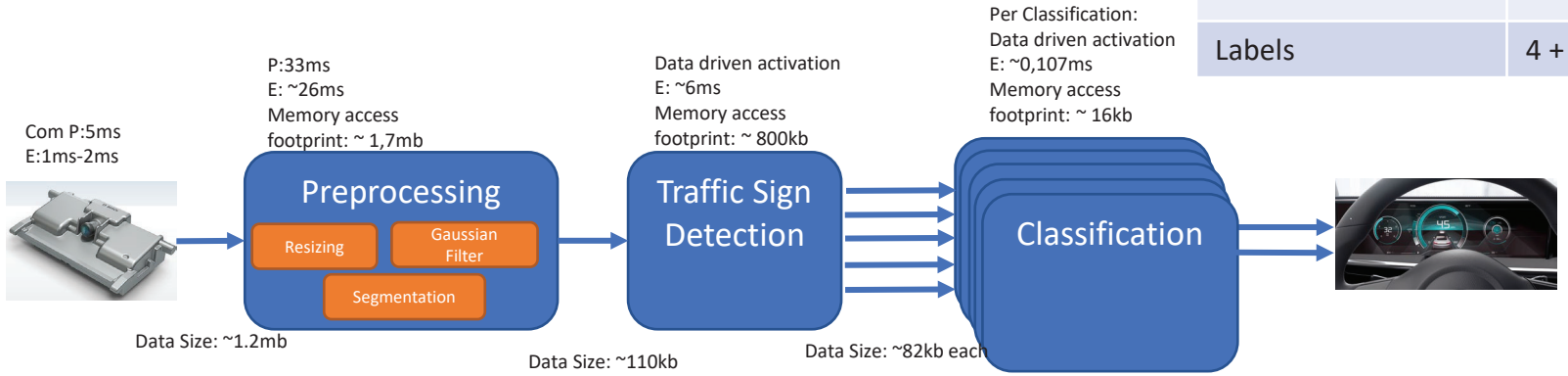


Traffic Sign Recognition (TSR)

- **Video processing & machine learning workloads**
  - Data-driven activation
  - Executed on CPU with offloading of computationally expensive calculations to GPU

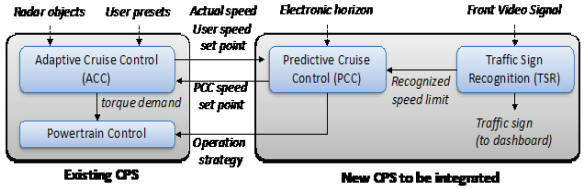


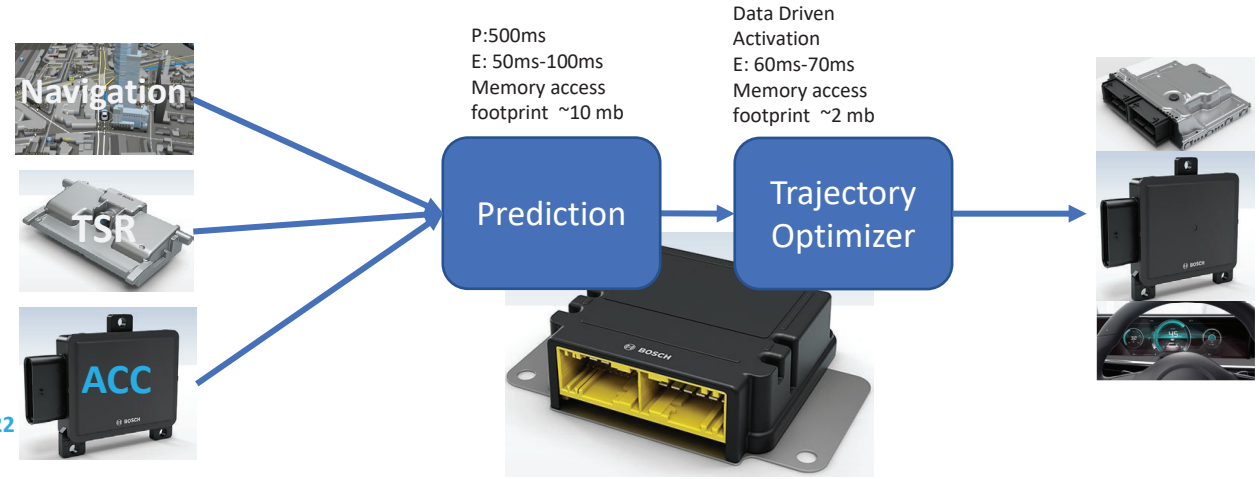
Amalthea Element	#
Tasks	3 + x (classification)
Runnables	7 + 3 * x
Labels	4 + x



Predictive Cruise Control (PCC)

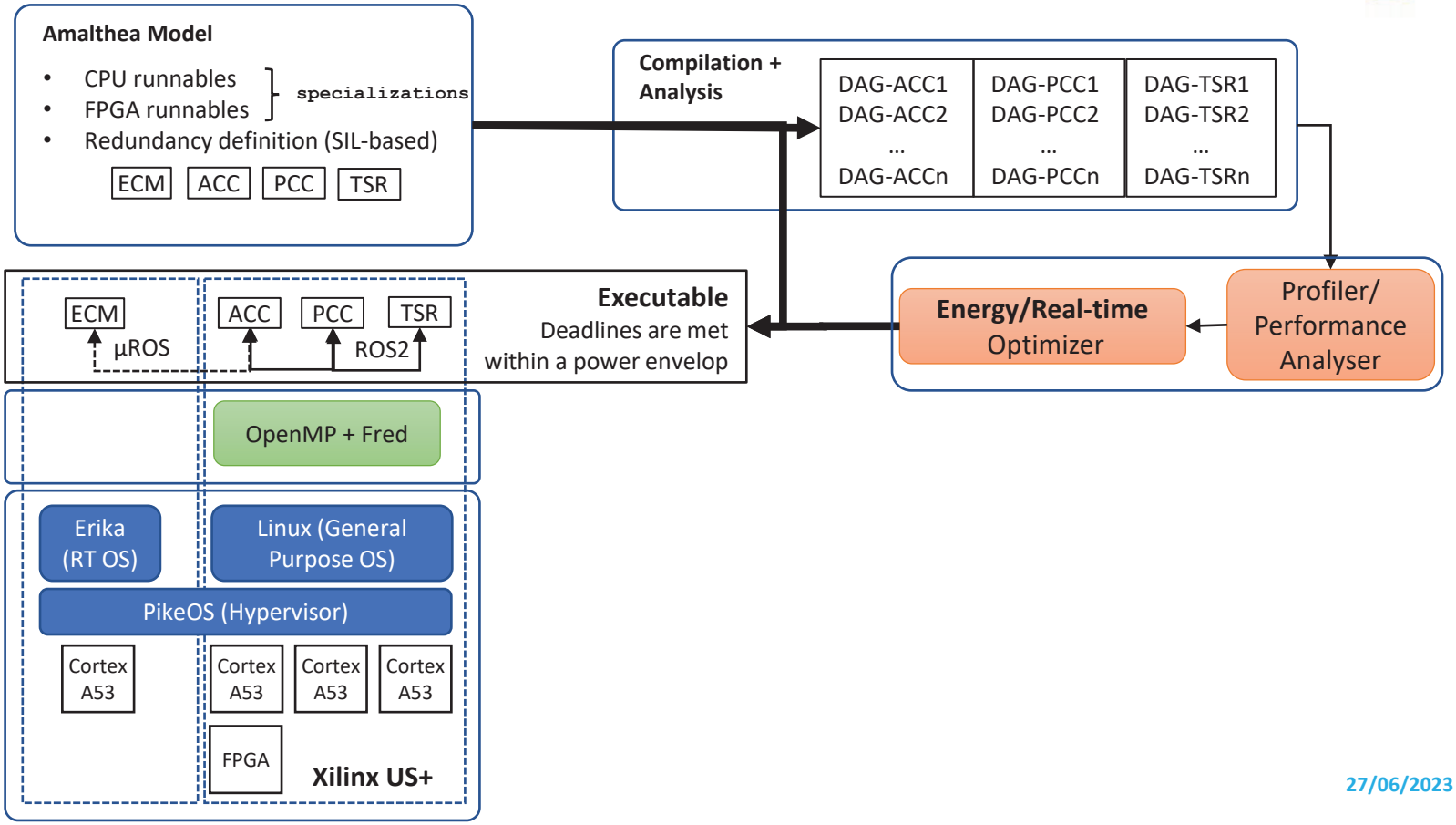
- Cyclic calculation (500 ms) of “electric horizon” for efficient driving mode
  - Based on map data and planned route (navigation system)



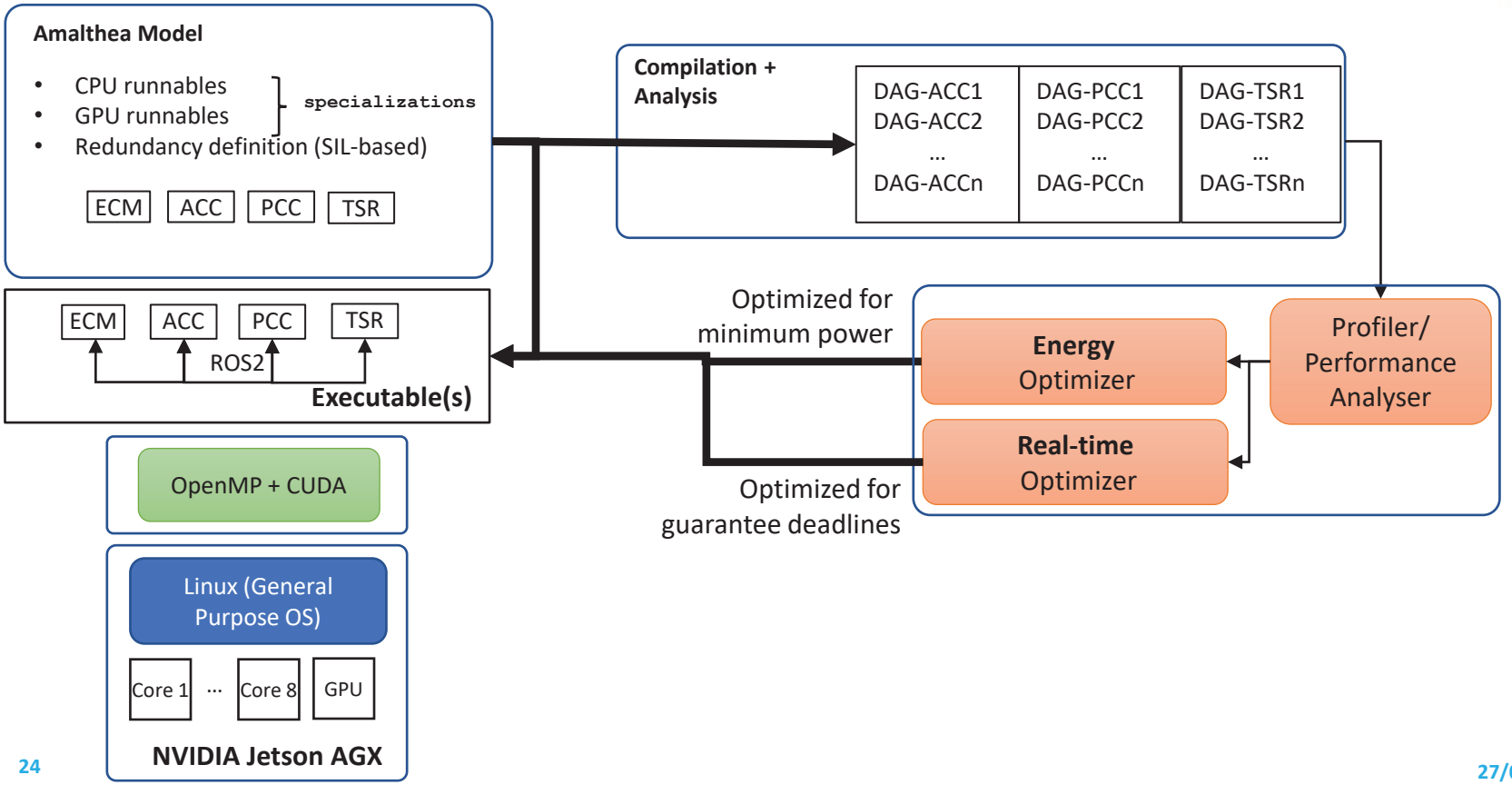


Amalthea Element	#
Tasks	2
Runnables	~10
Labels	~50

Evaluation: Xilinx Ultra-Scale+



Evaluation NVIDIA Jetson AGX



Thank you!

www.ampere-euproject.eu

The AMPERE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871669