HERCULES

High-Performance Real-time Architectures for Low-Power Embedded Systems

HERCULES: SOFTWARE ARCHITECTURES FOR NEXT-GENERATION AUTONOMOUS DRIVING PLATFORMS

PAOLO GAI, EVIDENCE SRL, ITALY
MARKO BERTOGNA, UNIVERSITY OF MODENA, ITALY
VALERIO GIORGETTA, MAGNETI MARELLI, ITALY
FULVIO TAGLIABÒ, MAGNETI MARELLI, ITALY
DAVIDE FONTANA, MAGNETI MARELLI, ITALY
MICHAL SOJKA, CTU PRAGUE, CZECH REPUBLIC

14TH AUTOMOTIVE SPIN WORKSHOP, MILAN 10 NOV, 2016

Agenda

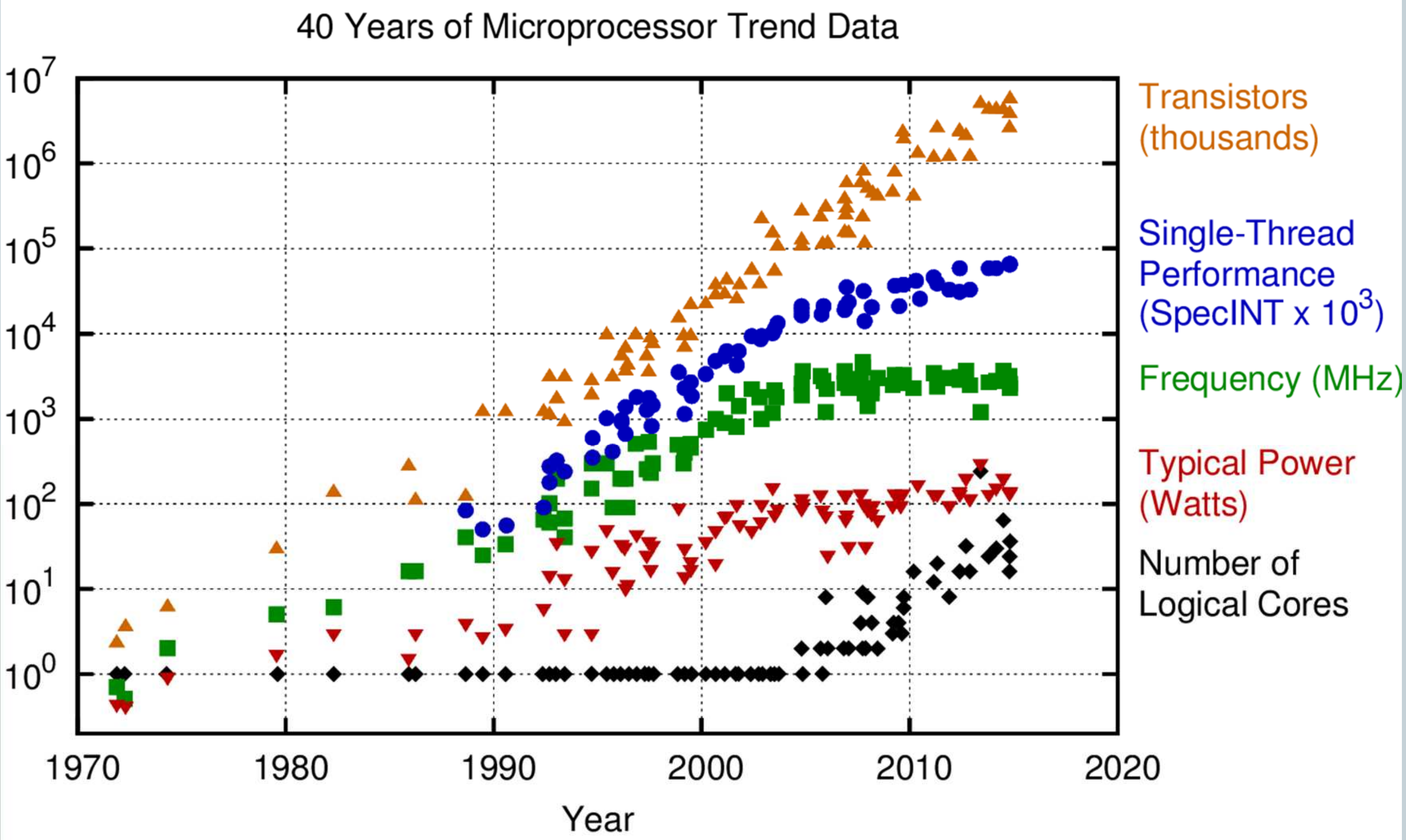
- Autonomous driving HW/SW application trends
- HERCULES goals and use cases
- HW and SW architecture used in the project
- Predictability issues, MemGuard and PREM
- ISO26262 implications

Paolo Gai – pj@evidence.eu.com

Applications Trend

- New applications requiring a prompt interaction with the environment
- Replace human activities
  - Driving, flying, sailing, farming, tracking, manufacturing, building, checking, testing, etc.
- Higher workload
  - E.g., from multiple cameras and sensing devices
  - Require parallel computing platforms/accelerators
- Real-time guarantees
  - What if a self-driving car "misses" a deadline?
- Higher criticality/safety requirements

Technological trend



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Observations

- You will be using multi/many-core systems
- Performance will keep growing only for properly designed parallel applications
  - Deep learning and Convolutional Neural Networks
- Mastering parallelism is not so easy
- Achieving a predictable behavior is harder
  - Parallel concurrency: inter-core dependencies
  - Multiple contention sources: bus, caches, memory, I/O, etc.
- Existing solutions either sacrifice performance (overprovisioning) or predictability
- What about Safety?

HERCULES target

- Real-Time Embedded Super-Computing Platforms
- Integrated framework to achieve predictable performance on top of cutting-edge heterogeneous COTS multi-core platforms
- Technological baseline
  - Real-time scheduling techniques and execution models recently proposed in the research community
  - High-performance/Low-power embedded COTS platforms
  - Next generation real-time applications

Paolo Gai – pj@evidence.eu.com

Main Goals

- Goal G1
  - Demonstrate and implement the first **industrial-grade framework** to provide **real-time guarantees** on top of **cutting-edge heterogeneous COTS platforms** for the embedded domain
- Goal G2
  - Obtain an **order-of-magnitude improvement** in the **energy efficiency** and **cost** of **next generation real-time systems**
- Goal G3
  - Provide a **homogeneous programming interface** to simplify the development of future real-time application on top of heterogeneous COTS platforms

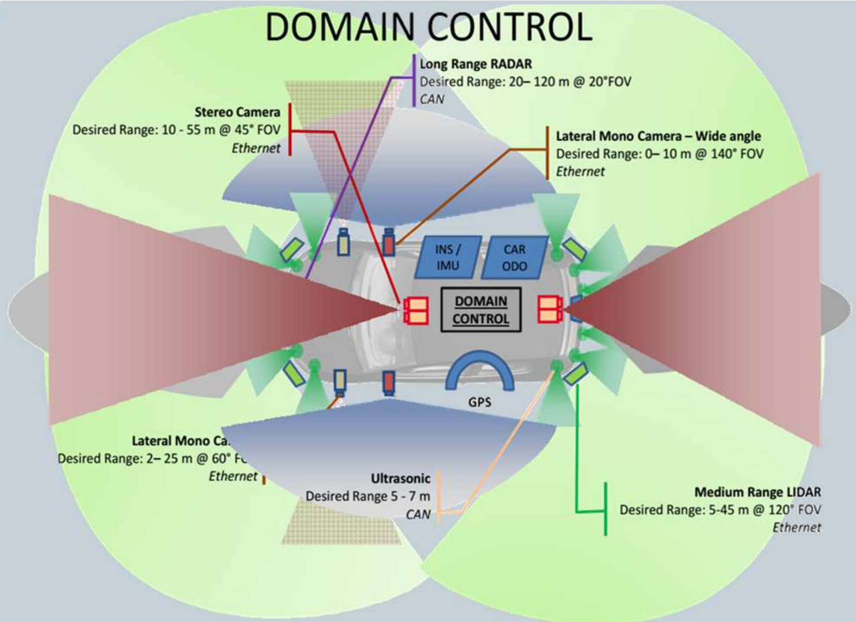
Paolo Gai – pj@evidence.eu.com

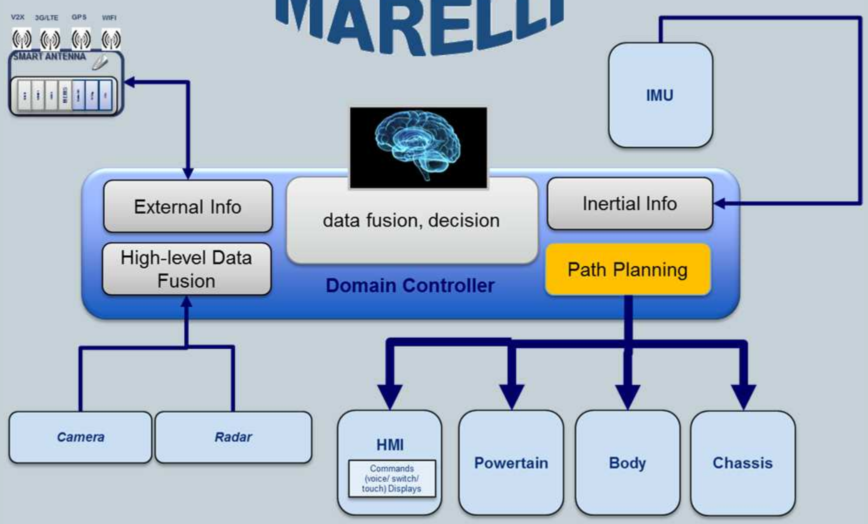
…a few questions…

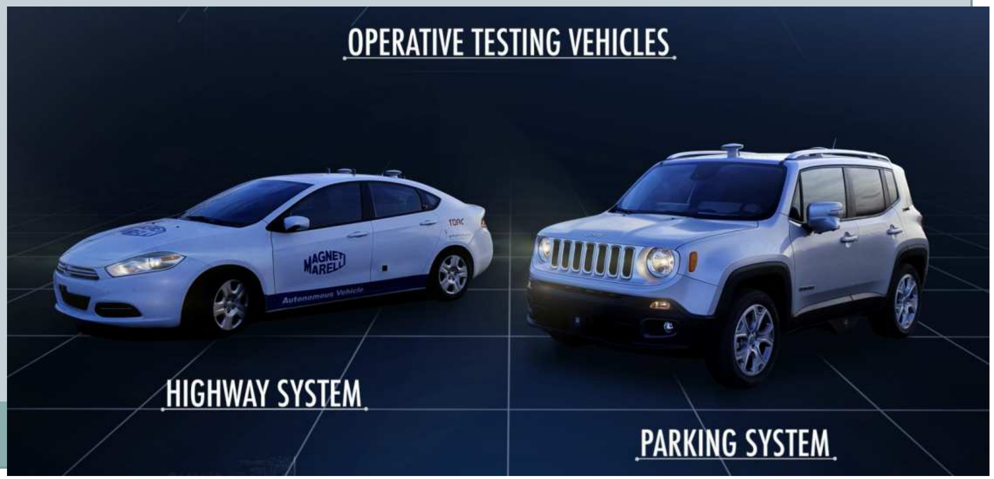
- What happens to the real-time predictability?
- How to integrate parallel programming models which are far away from the traditional static approach of AUTOSAR?
- How can we guarantee the ISO26262 Freedom From Interference?

Use Case 1: Autonomous Driving

- Domain controller
  - Multi-sensory data fusion
  - Situation awareness
  - Trajectory planning







Use Case 2: Avionics

AIRBUS GROUP

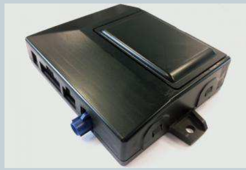
- Machine vision
  - Online computer learning for object detection and tracking

Onboard smart cameras (e.g. top: gearbox camera) provide visual feedback to pilots; their embedded processors are currently only used for simple image enhancement tasks. Hercules could allow Airbus to port its more complex machine learning and computer vision algorithms to these low-power platforms and offer additional features, e.g. surveillance while the airplane is parked on the airport apron (right).

HERCULES at a glance

http://image-sensors-world.blogspot.it/2015/07/ambarella-acquires-automotive-vision.html

- Cost
- Power
- Size



+Isolation
+Predictability
+Programmability
+Safety
+Openness

Paolo Gai – pj@evidence.eu.com

Hardware Platform

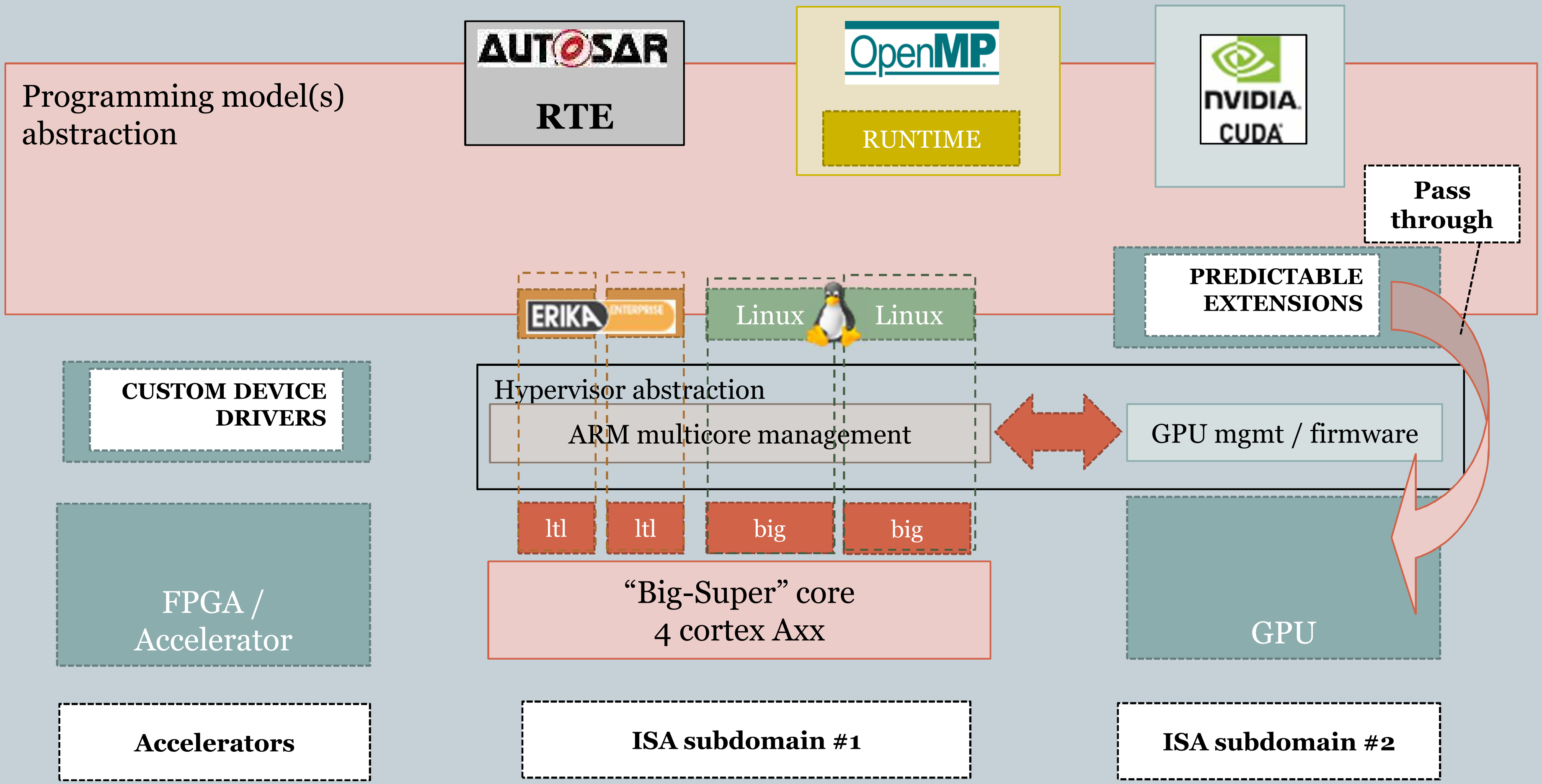
- Multi-core host + accelerator(s)
  - ARM big.little or similar power-efficient multi-core host
  - GPU, DSP cluster, many-core fabric or FPGA acceleration
- Two representative platforms selected
  - Nvidia Tegra Parker (16nm)
  - Xilinx Zinq Ultrascale+ (16nm)

other platforms evaluaterd were:

- Renesas R-Car H3 (16nm)
- Samsung Exynos 7 Octa (14nm)
- Qualcomm Snapdragon 820 (16nm)
- Intel 5th gen Core (14nm)
- Kalray MPPA (28 nm)
- TI KeystoneII (28nm)

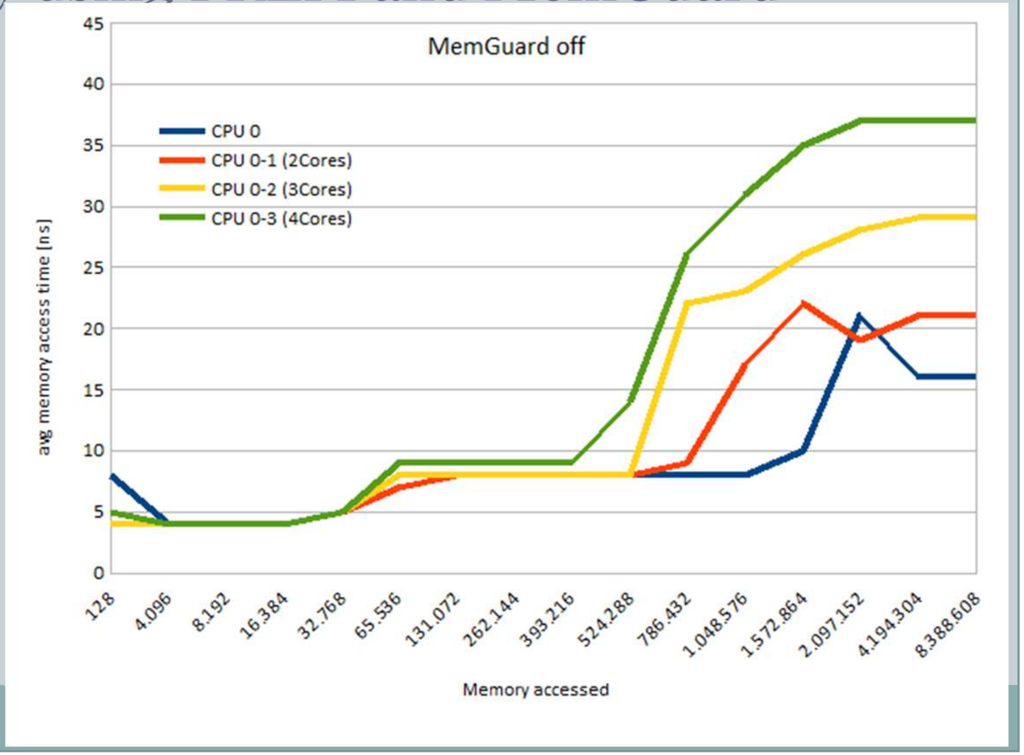
Paolo Gai – pj@evidence.eu.com

HERCULES Software architecture



Software Platform: Hypervisor

- Predictable Hypervisor
  - Based on NVidia Vibrante and on JailHouse
  - 10k LOC → Certifiable!
  - Predictability enhanced by using PREM and MemGuard techniques
- Memory and multicores
  - Effect on the execution time on multicores

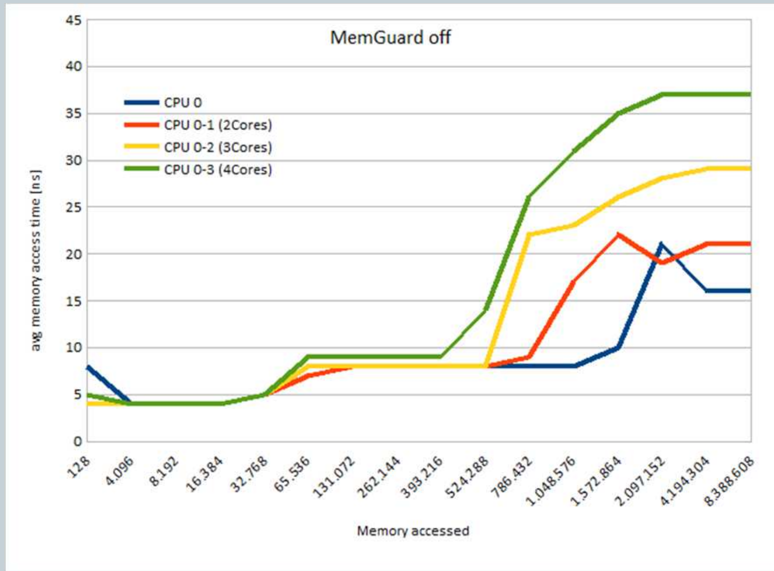


What about
ISO26262
Freedom From Interference?

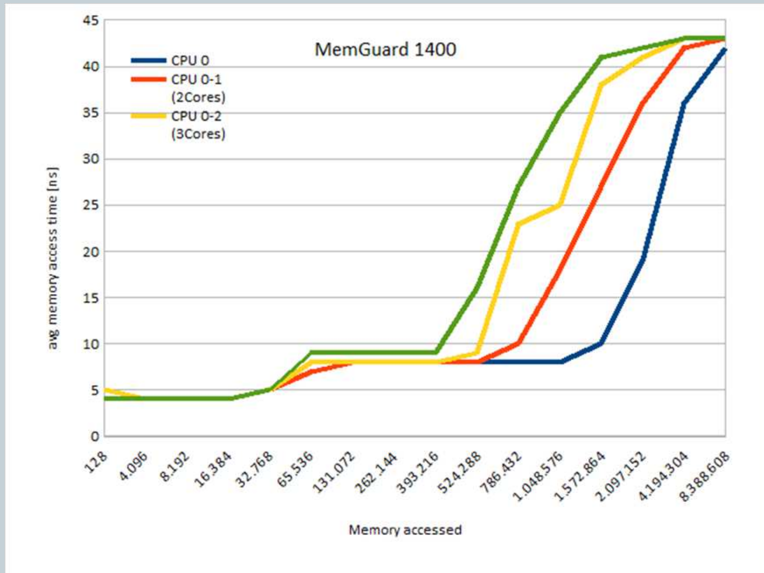
Software Platforms - MemGuard

- MemGuard - Main idea
  - Use CPU performance counters to limit the memory access of misbehaving cores/applications

Before MemGuard



After MemGuard



Paolo Gai – pj@evidence.eu.com

Software Plaftorm: Operating systems

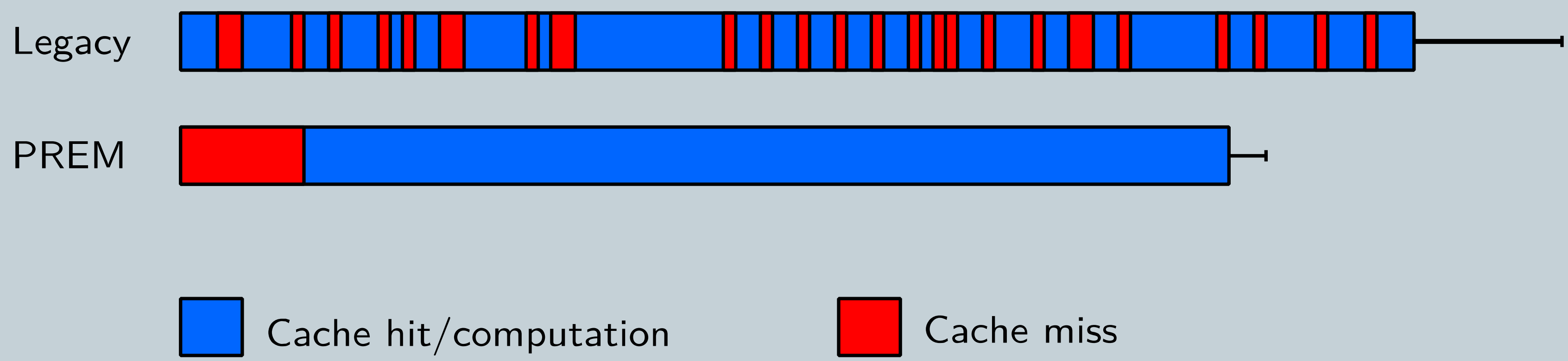
- Partitioning of RTOS into cores
  - One OS per core
- AUTOSAR subsystem
  - ERIKA ENTERPRISE http://erika.tuxfamily.org open-source and Made in Italy!
  - On the Cortex A5x and on the Cortex R5
- General Purpose OS
  - Linux on the Cortex A5x
  - SCHED_DEADLINE extended to Energy Aware Scheduling
  - Additional scheduling algorithms (GRUP, MBROE)

Paolo Gai – pj@evidence.eu.com

Software Platforms: Programming models

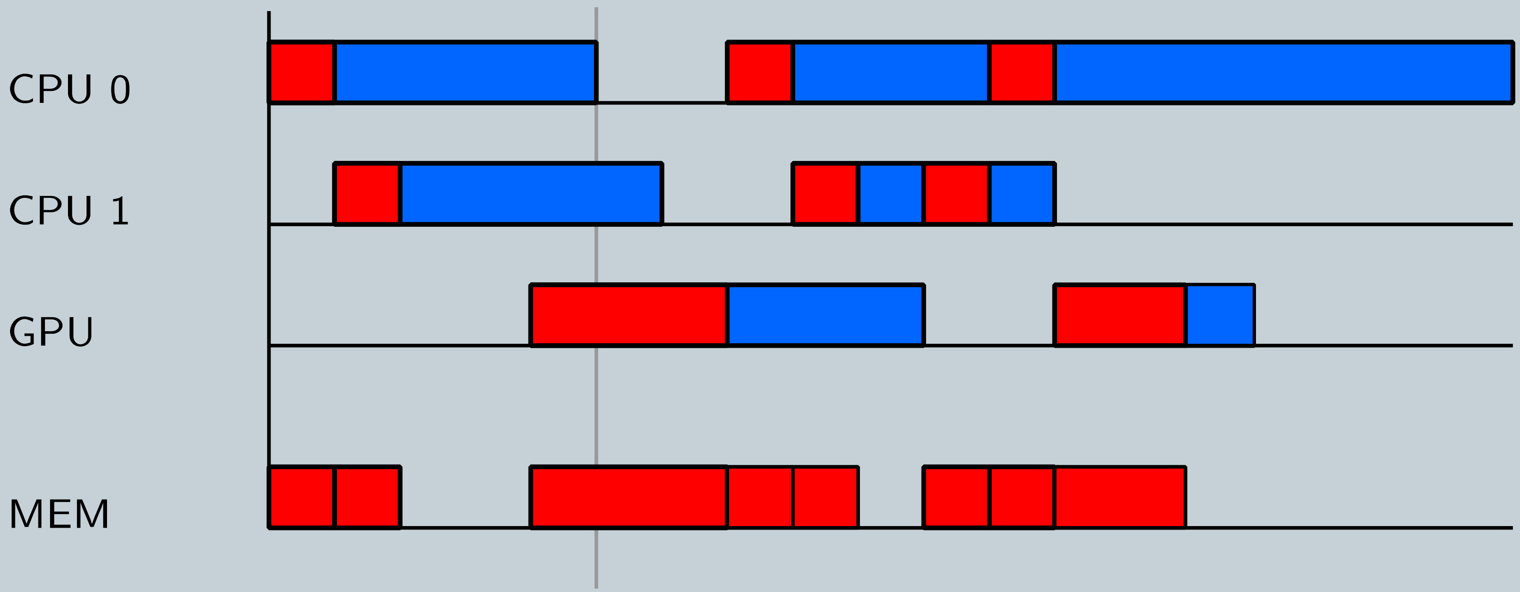
- We will host two main programming models
- AUTOSAR subsystem
  - RTE generator for a subset of the RTE specification
- Linux + GPU subsystem
  - Lightweight OpenMP with additional support for CUDA
- Integrated communication between AUTOSAR and Linux/OpenMP
- Support for PREM models
  - Compiler level modifications

Predictable Execution Model (PREM)



- Predictable interval
  - Memory prefetching in the first phase
  - No cache misses in the computation phase
  - Non-preemptive execution

PREM scheduling



- Memory-phase = Multi-resource activity
- CPU-centric scheduling is not sufficient
- Resource Constraint Project Scheduling

ISO26262 and Freedom From Interference

Finally, a few comments on ISO26262:

- Multi- Many- core HW architectures currently aim for ASIL B/C (not ASIL D!)
- Small Hypervisors (10k LOC)
  - Potentially Certifiable
- Freedom From Interference is not trivial
  - Predictability is not «just» obtained using an Hypervisor!
  - You need more advanced techniques like PREM and MemGuard

Paolo Gai – pj@evidence.eu.com

Conclusions

- HERCULES will provide a **software framework** to simplify the development of **next-generation real-time applications** on **heterogeneous COTS platforms**

- Multiple targets:
  - Performance with real-time guarantees
  - Low power/Low cost
- Mostly open-source
  - Linux, ERIKA, OpenMP
- Support for future Autonomous driving scenarios
- ISO26262 taken into account for future certifications

Paolo Gai – pj@evidence.eu.com

Hercules Project Partners

- Partners

1 (Coordinator)	University of Modena	UNIMORE	Italy
2	Czech Technical University in Prague	CTU	Czech Republic
3	ETH Zurich	ETHZ	Switzerland
4	Evidence Srl	EVI	Italy
5	Pitom snc	PIT	Italy
6	Airbus Gmbh	AB	Germany
7	Magneti Marelli	MM	Italy

- Timespan
  - January 2016 – December 2018
- Budget: ~3.3 M
  - 2.1M EU, 700k Switzerland, 500k industrial co-funding
- Website
  - http://hercules2020.eu/