



Challenges of FSW Schedulability on Multicore Processors

Flight Software Workshop
27-29 October 2015

Marek Prochazka
European Space Agency

MULTICORES: WHAT DOES FLIGHT SOFTWARE ENGINEER NEED?

- ❑ **Space-qualified multicore processor (obviously)**
- ❑ Case studies
- ❑ Operating system
- ❑ Compiler
- ❑ Emulator
- ❑ Other tools
 - Parallelization
 - Testing/Debugging
- ❑ Scheduling approach
- ❑ Timing analysis
- ❑ Benchmarks
- ❑ Demonstrate technology with existing flight SW
- ❑ ...

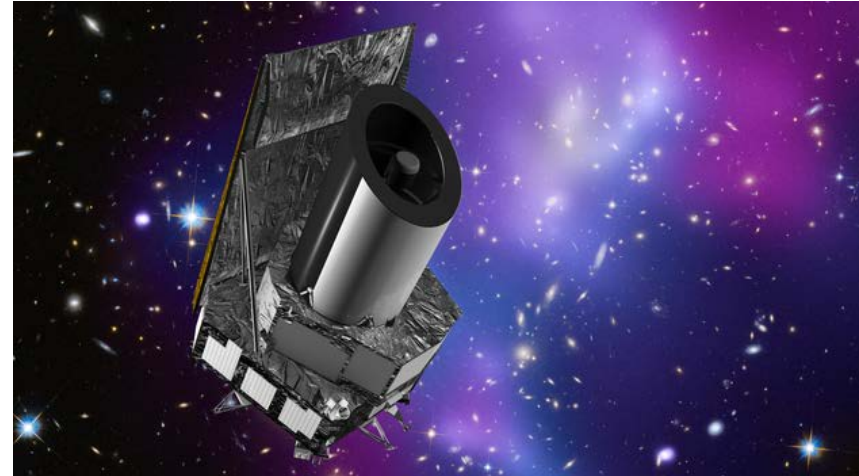


CASE STUDIES FOR MULTICORES IN SPACE (I)



□ Data processing for Euclid

- Nature of dark matter and dark energy by accurately measuring acceleration of universe
- Launch 2020, L2 orbit, 7 years mission
- 1.2 m telescope with H2RG state-of-the-art infrared detectors
- Usually data processing done on ground but not for Euclid:
 - L2 satellite
 - Efficiency of observation (downlink during 4 hours/day)
- 16 detectors, 2048*2048 pixels per detector
- For each detector and each frame: multi parallel operations (bias, reference pixels correction)
- Non-optimized results with LEON2 show too much time needed
- **Large focal plane needs multi-core processing**

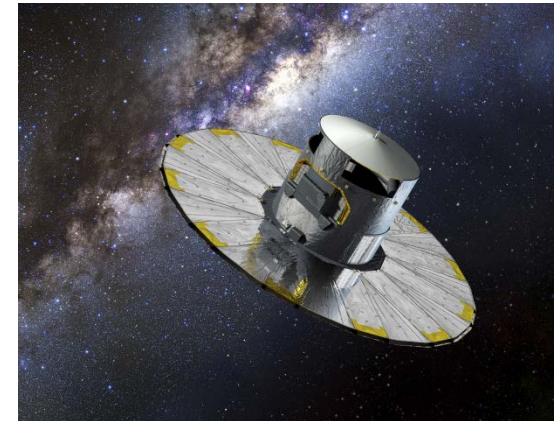


CASE STUDIES FOR MULTICORES IN SPACE (II)



□ Gaia VPU demonstration on LEON4-NGMP

- Billion stars three-dimensional map of our galaxy
- Launched 2013
- **RTEMS SMP for LEON3/LEON4 (ESA activity)**
- Porting Video Processing Unit (VPU) code to LEON4-NGMP and LEON3-GR712
- Parallelizing the VPU application
 - **MTAPI by Multicore Association (MCA)**
- Speed-up 2.6 from single core to 4 cores



□ Advanced GNC needing multicores

- Intelligent image processing for entry descent and landing
- Deorbiting uncooperative flying objects

□ Proba FSW demonstration

- Porting Proba Data Handling software and image processing



- ❑ **Space qualified GR712RC Dual-Core LEON3FT SPARC V8 Processor**
- ❑ **ESA with Cobham Gaisler developing LEON4-NGMP/GR740 processor**
 - Fault-tolerant quad-core SPARC V8 integer unit with 7-stage pipeline, 8 register windows, 4x4 KiB instruction and 4x4 KiB data caches
 - System frequency: 400 MHz (TBD)
 - Two double precision IEEE-754 FPUs shared between pairs of cores
 - 128-bit Processor and Memory AHB bus
 - MMU and L1 cache per core
 - 256 KiB shared L2 cache
 - ...
- ❑ **LEON4-NGMP Presented by Cobham Gaisler at FSW 2012 including benchmark results**
 - Branded as GR740

- **Early study assessing multicores for ESA spacecraft**
 - Analyze system level impact of multicore processors use in European space missions
 - NGMP/GR740 assessment
 - Identification of adapted software techniques
 - Execution models
 - Task distribution and synchronization
 - I/O management
 - Software tools (OpenMP, MPI for parallelization)
 - Guidelines for multicore use in space applications
 - Integrated Modular Avionics (XtratuM on NGMP/GR740)
 - Optimizations for onboard data processing

❑ Issues with multicores

- Most software components are inherently sequential and therefore not suitable for parallelization
- Parallelization implies complexity at software design level
 - Synchronization, deadlock and starvation avoidance
 - Etc.
- Shared resources imply interference
 - Potentially huge impact on software behavior
 - Could break independence between different software modules



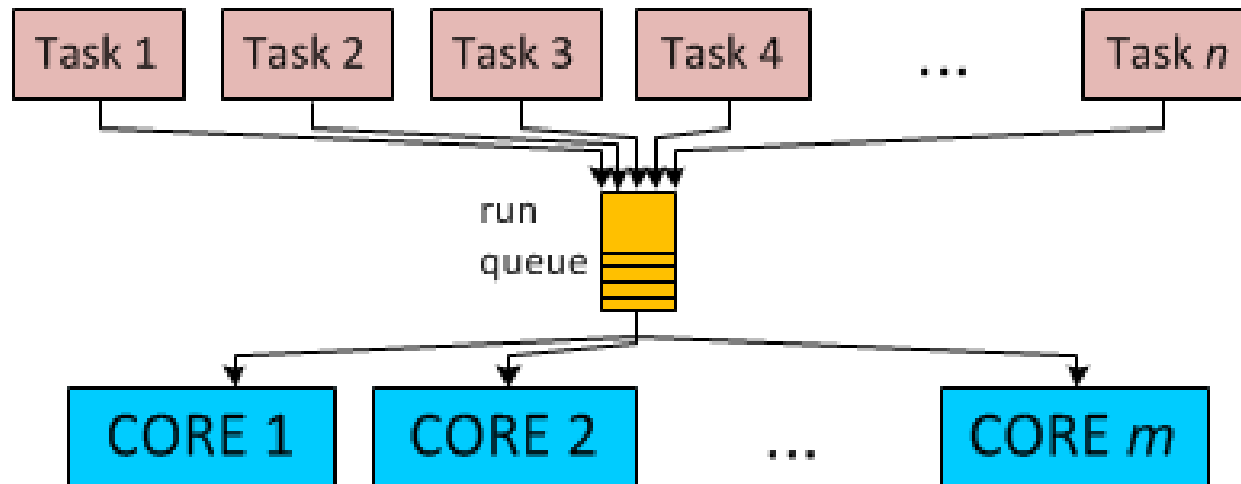
- ❑ **Classical approach to FSW schedulability on single-cores**
 - Based on Worst-Case Execution Time per task (WCET)
 - Fixed priorities
 - Response-time analysis or Rate-monotonic analysis
- ❑ **Classical approach is not possible for multicores**
 - Multiple tasks execute at the same time (one per core)
 - WCET harder to analyse due to inter-task interferences accessing shared resources
 - **It is hard to provide a safe and tight WCET estimation in multi-cores**
 - Arbitration mechanism
 - WCET depends on workload!
 - Scheduling tasks on multiple cores is more complex

- ❑ **Key requirement: Time Composability**
 - WCET computed for a task in isolation is not affected by other software in the system
 - Enables incremental qualification and system upgrades

- ❑ **Different types of scheduling**

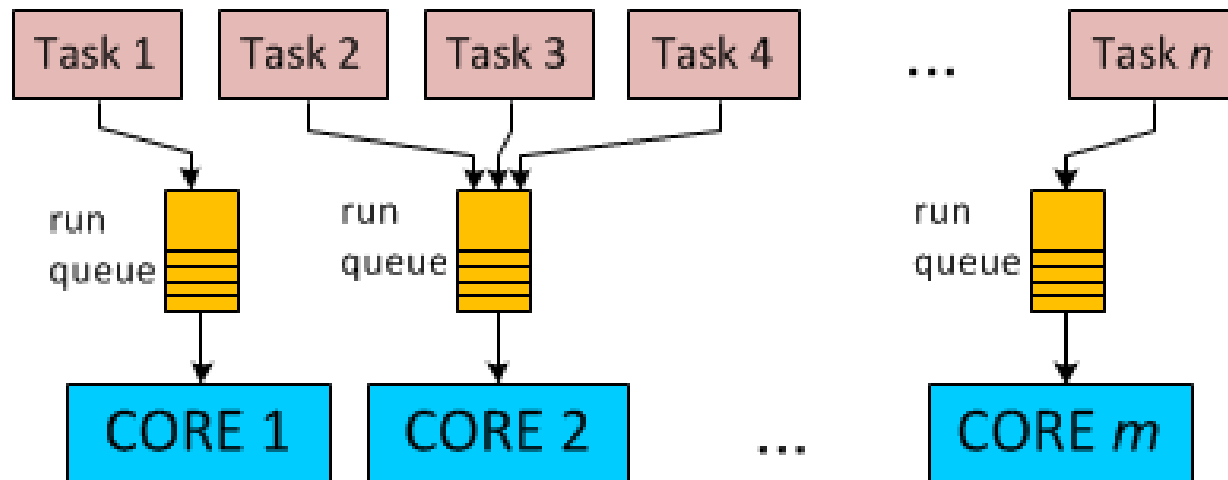
□ Global scheduling

- Dynamic task binding
- Single scheduler, single run queue
- Better utilization of all cores
- Overhead of task migration



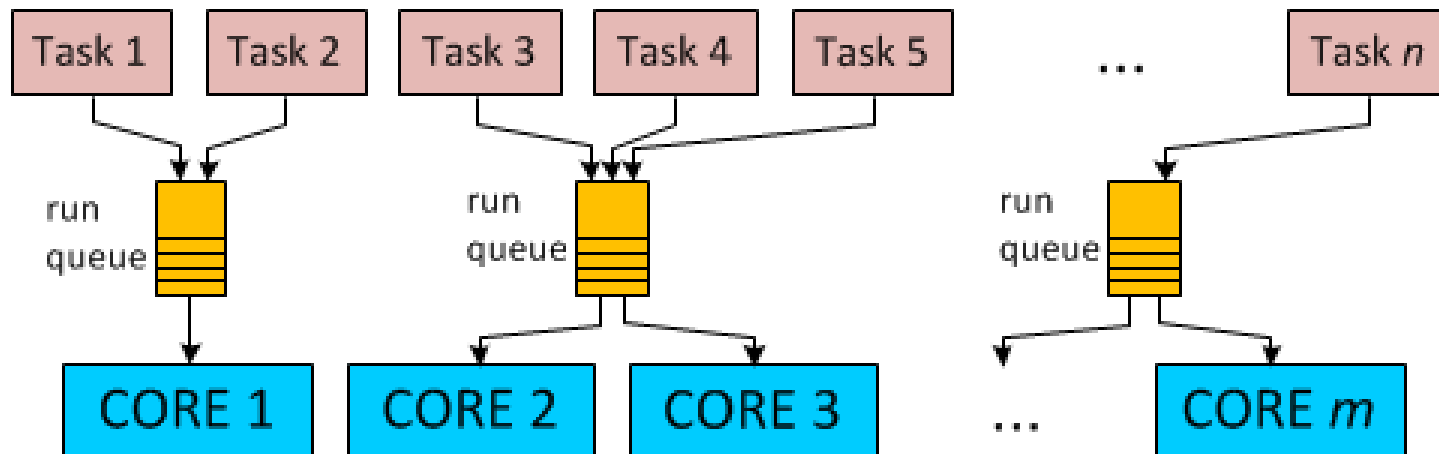
□ Partitioned scheduling

- Static task binding
- Each core with its own scheduler, its own run queue
- Lower utilization
- Better average response times



□ Hybrid

- Single scheduler/run queue per pre-defined number of cores
 - Could also be one queue per core
- Statically configurable



❑ Designed a benchmark suite for multicores

- Suitable to exercise quad-core GR740
- Capable of generating different inter-task interference scenarios that may arise in GR740
- Executed on
 - Xilinx ML510 development board implementing GR740 in its FPGA (quad-core)
 - GR712RC (LEON3 dual-core)

❑ Main goals:

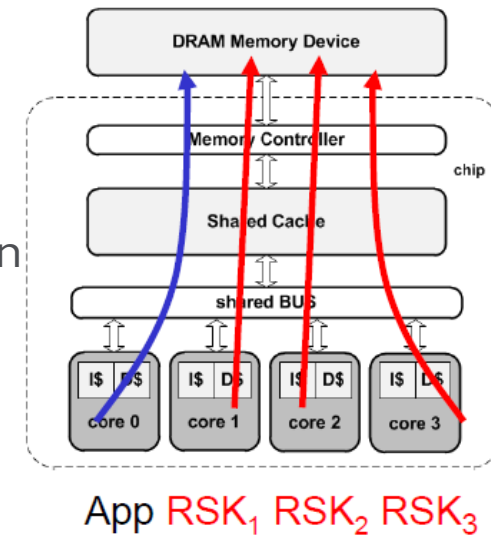
- Understanding how inter-task interferences affect performance and predictability
- Understanding of how to stress GR740 resources and how proposed benchmarks mimic ESA reference applications

❑ Microbenchmarks aka Resource Stressing Kernels (RSK)

- Single-behavior kernels that constantly access a shared resource
 - Put high pressure on that resource (bus, memory, cache)
- Used as co-runners to determine the slowdown a given application may suffer due to conflicts in that resource

❑ Observations

- Inter-task interferences have a significant impact on observed execution times on COTS multicores
- NGMP/GR740 observed slowdown due to inter-task interferences is higher than for the GR712RC
 - Higher number of cores
 - Inclusion of a shared L2 cache



- ❑ **CPU intensive tasks:** Little effect observed due to inter-task interference
- ❑ **Memory intensive tasks with no store instructions:** Up to 4.3x slowdown depending on the level of inter-task interference:
 - 83% if interference is only in the AMBA AHB processor bus
 - 2.6x if interference is in AMBA AHB processor and memory buses and memory controller
 - 4.3x if interference is in the AMBA AHB processor and memory buses, L2 cache, and memory controller
- ❑ **Memory intensive tasks with many store instructions:**
 - Up to 20x slowdown, depending on the utilization of L2 and the AMBA AHB bus

- ❑ **Note: Linux and RTEMS SMP AHEAD version used**

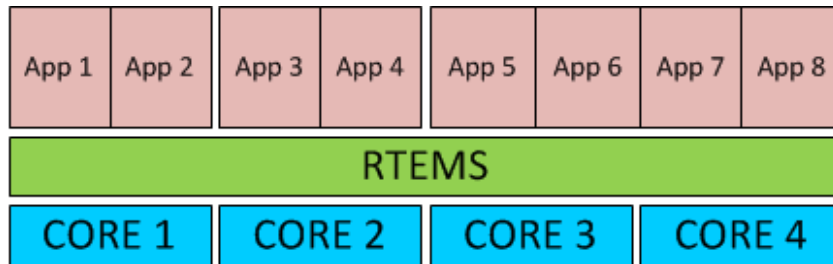
□ Challenges:

- SW level: Impact on task allocation and scheduling
- HW level: HW-support for inter-task interferences

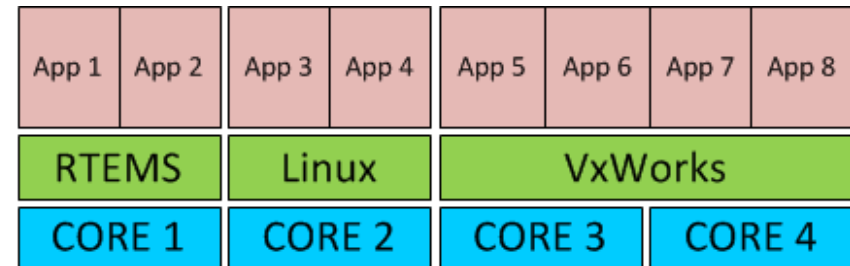
MULTIMA: Multi-core in Integrated Modular Avionics



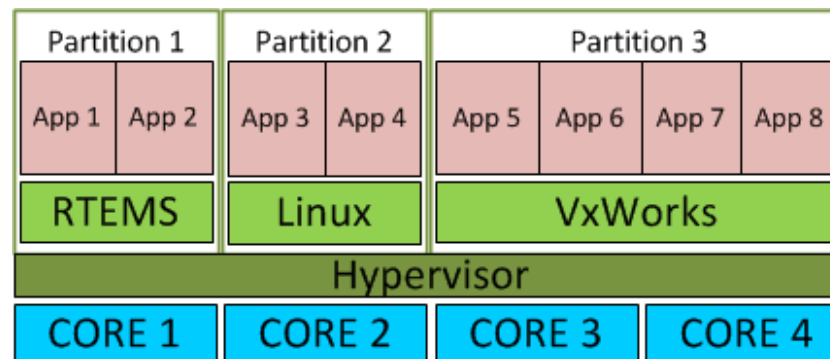
□ Symmetric multiprocessing



□ Asymmetric multiprocessing



□ Asymmetric multiprocessing with separation kernel/hypervisor



- ❑ **ARINC 653 partition scheduling on multiple cores**
 - Different partitions run concurrently on different cores
 - Partition operating system does not have to support multicore
- ❑ **No need to parallelise applications**
- ❑ **Concurrently executing partitions suffer from interference**
 - Distributed IMA works on general multiprocessors but not on multicores as designed today
 - Use PMCs to monitor/control interference



□ Use partitioned scheduling for hard real-time systems (platform)

- Slightly lower utilization
- Avoid task migration and synchronization across cores



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

□ Partial time-composability

- Use fully time-composable WCET if available
- Try to find a partially time-composable WCET if executing all workloads is feasible
- Optimize task allocation per core
- Note: This approach is feasible only with small number of cores

□ Use probabilistic WCET if needed

- Perform many measurements with Resource Stressing Kernels running on other cores
- Estimate probability of overrun (Extreme Value Theory)

□ Scheduler per core with no assumptions on workload on other cores

- ❑ **Solutions based on hardware features**
- ❑ **Main goals:**
 - Ease the adoption of multicore processors by the European Space Agency
 - Analyze and improve on-chip shared resources in terms of time predictability and time composability in deterministic and time-probabilistic architectures
- ❑ **Several proposals developed to ease computation of WCET estimates for multicores**
 - Either by means of removing interactions between tasks or
 - Upper-bounding interaction between tasks
 - Objective: Creating hardware support for taking inter-task interferences into account when computing WCET estimations for the NGMP

□ Performance Monitoring Counters (PMC)

- Provided by GR740 to enable run-time information collection linked to certain events, e.g.
 - Data and instruction cache misses
 - L2 cache misses
 - Total number of executed instructions
 - Number of memory operations
 - Number of executed cycles
 - Processor AMBA bus usage

□ It is proposed to add a PMC indicating interferences

- Contention prediction model based on PMCs
- PMC(s) to measure actual contention
- Could be used as a guideline for other applications
- Could be monitored and trigger recovery (e.g. by killing a task which is deviating from its guideline)
- Use for testing/debugging

Architectural Solutions for Timing Predictability of NGMP (III)



- ❑ **Arbitration policy on the bus**
 - Round robin vs. TDMA
 - Round robin seems better because in TDMA slots stay unused

- ❑ **Partitioned L2 cache with adequate support by the AHB AMBA bus processor**

❑ QERx

- Instruction-level emulator of ERC32 and LEON processors
 - Built upon QEMU open-source dynamic translation emulator
 - Based on block translation
 - Not instruction accurate
- Faster than real-time
- For multicores traditional emulation unfeasible, emulator architecture change required
- Currently emulates Dual-core LEON3 (up to 8 times faster)
- Ready for LEON4-NGMP/GR740

❑ **GR740 emulator able to emulate HW interferences from other cores**

❑ **LLVM compiler optimizations for multicore**

- ❑ **Presented some challenges of multicores onboard spacecraft**
- ❑ **Hardware**
 - NGMP/GR740 characteristics
 - Performance Monitoring Counters
- ❑ **Software tools**
 - RTEMS SMP for GR740
 - Benchmarks
 - Compilers (LLVM optimizations for multicores)
 - Emulators
 - QERx
 - GR740 emulator able to emulate HW interferences from other cores
- ❑ **Scheduling approach**
 - Partial time-composability
 - Probabilistic timing analysis

CONTRIBUTORS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



THANK YOU

**Presenter: Marek Prochazka
European Space Agency**

**Main ESA contributors: Marco Zulianello, Luca Fossati, Jorge Lopez
Contact: First.Last at ESA.int**

**This presentation contains material delivered in the scope of
several ESA activities**



BACKUP SLIDES

❑ Multicores are widely used in home/business

- Desktop computers, laptops, tablets and phones
- Embedded devices (network processing, digital signal processing)

❑ Pros

- Solution to high power consumption of processors with high CPU frequency
- Simple core design
- Mixed-criticality applications
 - Hardware utilization is maximized, while cost, size, weight and power requirements are reduced
- Parallelization of computations
- Systems with limited space/power

□ Cons

- Shared resources between cores (bus, memory)
- Single core performance is usually lower
- Execution on multiple cores requires functional isolation
 - Prevent that one application corrupts the state of other applications
 - Low-criticality applications must not affect high-criticality ones
- Multicores usually not designed for real-time applications, but for data crunching
- **Harder to analyze and prove timeliness**
 - **It is hard to provide a safe and tight WCET estimation in multi-cores**
 - **Due to inter-task interferences via shared HW resources**

□ Multi-cores offer better performance per watt than single-core processors

- Expected technology trend also in time-critical systems

- ❑ **European project (FP7) with multiple partners**
 - Barcelona Supercomputing Center, Rapita, INRIA, Airbus, University of Padua
 - ESA one of the industrial advisors
- ❑ **Objective:** To define new hardware and software architecture paradigms that, by construction, exhibit a timing behaviour that can be analysed with probabilistic techniques
 - Define a new way of designing and analysing reliable software systems using probabilities in timing analysis
 - Moves from timing-deterministic systems towards timing-randomised systems that exhibit truly independent timing behaviour and therefore enable application of theory of extreme numbers to (probabilistically) predict the behaviour of extreme execution times (i.e. probability of overruns)

□ Benefits

- Derive safe and tight execution bounds, requirements on overrun rates proportional to their criticality
- Reduce the complexity and time required for timing analysis
- Reduce pessimism

□ Probabilistic analysis depends on appropriate hardware design

- HW allows obtaining probabilities on series on “independent” measurements (provide randomised execution times)
- Exploring software-only randomisation

□ <http://www.proartis-project.eu/>

MERASA: Multi-Core Execution of Hard Real-Time Applications Supporting Analysability



- ❑ **European project (FP7) with multiple partners** (finished 2014)
 - Barcelona Supercomputing Center, Rapita, Honeywell, University Of Augsburg, IRIT/Uni. Of Paul Sabatier)
 - ESA one of the industrial advisors
 - <http://www.merasa.org>

- ❑ **parMERASA** (Multi-Core Execution of **Parallelised** Hard Real-Time Applications Supporting Analysability (finished 2014)
 - Parallelisation of hard real-time programs in avionics, automotive and construction machinery
 - Targeting multi-/many-core systems with up to 64 cores
 - WCET verification and profiling tools
 - Timing analyzable many-core architecture
 - Contributions to standards and open source software
 - <http://www.parmerasa.eu/>

MultiPARTES: Multi-cores Partitioning for Trusted Embedded Systems



- ❑ **European project (FP7) with multiple partners** (finished 2014)
 - ESA one of the industrial advisors
- ❑ **Main goals:**
 - Support mixed criticality for trusted embedded systems based on multicore open-source virtualization
 - Analysed scheduling techniques of partitioned systems on multicore platforms
 - Use of the XtratuM hypervisor (University of Valencia)
- ❑ <http://www.multipartes.eu/>

PROXIMA: Probabilistic Real-Time Control of Mixed-Criticality Multicore and Manycore Systems



- ❑ **European project (FP7) with multiple partners**
 - E.g. Airbus, Cobham Gaisler, Sysgo, Rapita, Barcelona Supercomputing Center, University of York...
 - Started 2013
 - ESA one of the industrial advisors

- ❑ **Main goals:**
 - Software timing analysis using probabilistic analysis for many-core and multi-core critical real-time embedded systems
 - Enabling cost-effective verification of software timing analysis including WCET

- ❑ <http://www.proxima-project.eu/>

- ❑ **Instruction-level emulator of ERC32 and LEON processors**
 - Built upon QEMU open-source dynamic translation emulator
 - Based on block translation
 - Not instruction accurate
- ❑ **Faster than real-time**
 - **Past (ERC32):** Slow processors, emulation speed not a problem
 - **Current (LEON2):** A gap starting to show between processor speed and traditional emulation
 - **Near future (LEON3):** Traditional emulation frustratingly slow
 - **Medium future (LEON4):** Traditional emulation unfeasible, emulator architecture change required
- ❑ **Currently emulates Dual-core LEON3 (up to 8 times faster)**
- ❑ **Ready for LEON4-NGMP/GR740**