Interference Measurement and Mitigation Means on Multi-Core COTS processors

Contacts : sylvain.girbal@thalesgroup.com
jimmy.lerhun@thalesgroup.com

THALES
RESEARCH & TECHNOLOGY
CRITICAL EMBEDDED SYSTEM LABS

EMC²

Avionics: Context & Challenges

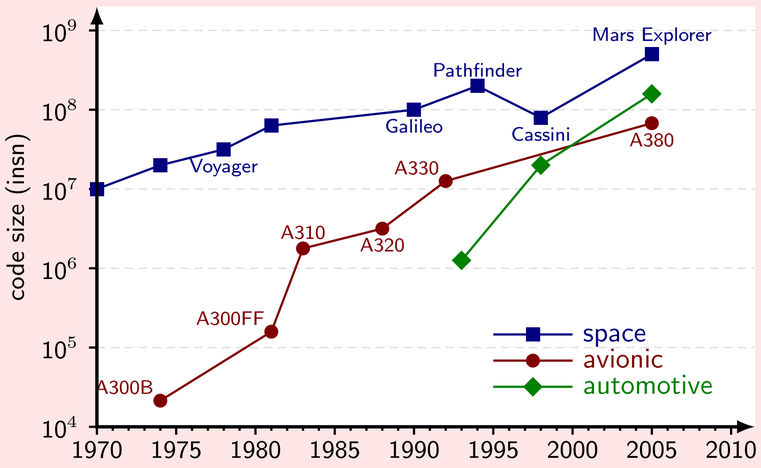
Requirements

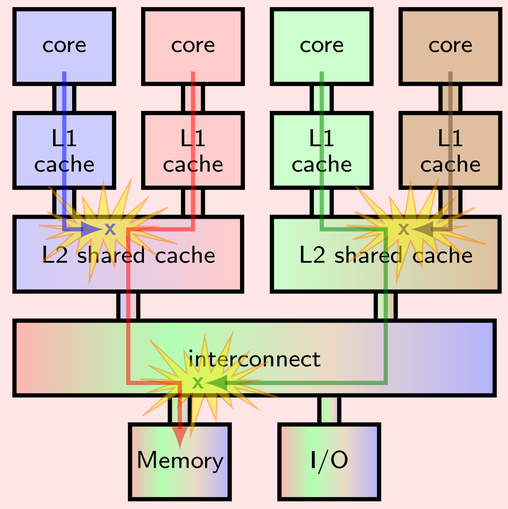
- Exponential increase of performance requirements
- Compliance with industry standards
- Spatial isolation & Timing isolation

The problem: inter-core interferences

- Multi-core COTS
- Timing interference while concurrently accessing the same shared hardware resource

How to evaluate solutions ensuring a time-deterministic usage of multicore architectures ?

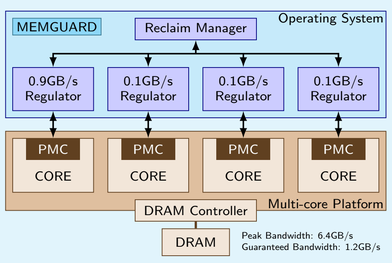




Deterministic Platform Software Solutions

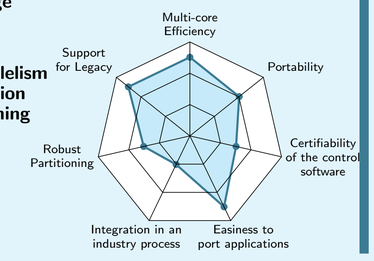
Regulation DPS: react before interferences become harmful

MemGuard

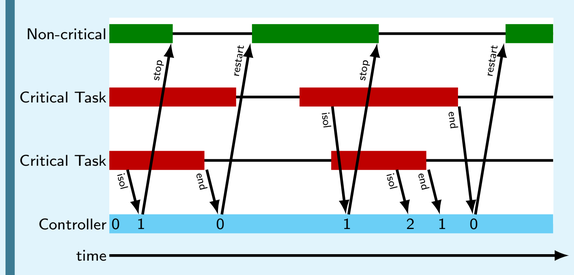


- Pre-allocated maximum bandwidth per timeslots
- Keeping interference within an acceptable range

- Support for legacy
- Fully exploits parallelism
- Bandwidth evaluation
- No robust partitioning

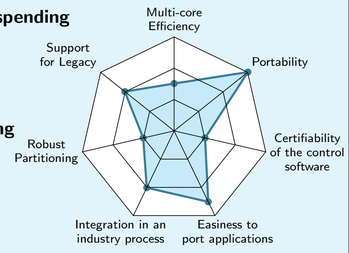


Distributed Runtime WCET Controller



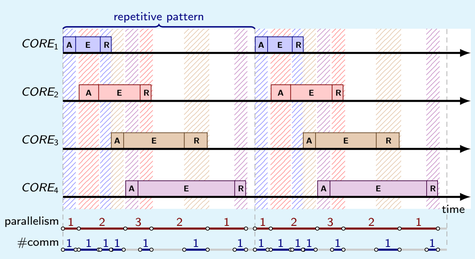
- Mixed critical context
- Regularly checks if critical tasks can tolerate interferences from other tasks
- WCET controller suspending non-critical tasks

- No critical legacy
- Easily portable
- No robust partitioning



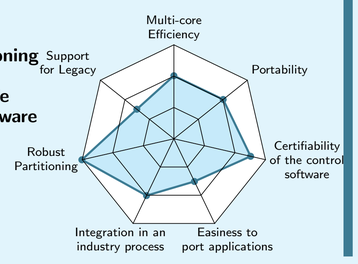
Control DPS: Avoid interferences by restricting usage

Deterministic Execution Models

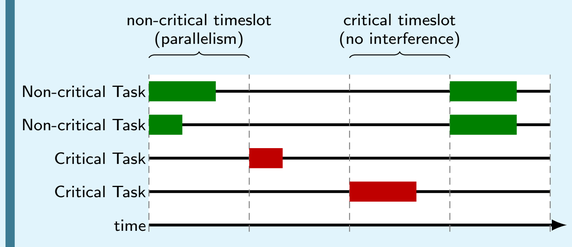


- Static schedule with:
- parallel execution phases
- sequential communication phases

- Ensures time partitioning
- Restricts parallelism
- Alter the source code
- Simple platform software

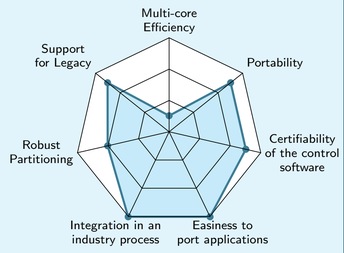


Deterministic Adaptive Scheduling

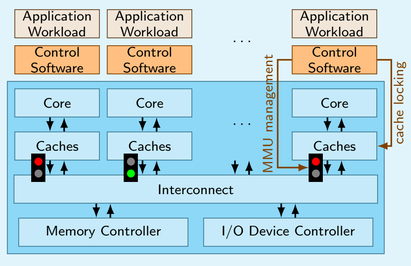


- Mixed critical context
- Standalone Critical and parallel non-critical time slots

- Restricts parallelism
- Already used in an industry process
- No modification to legacy software

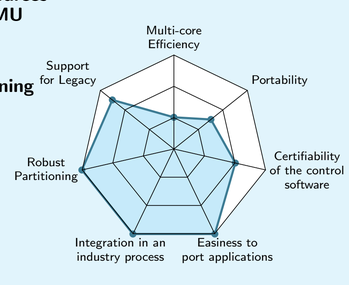


Marthy

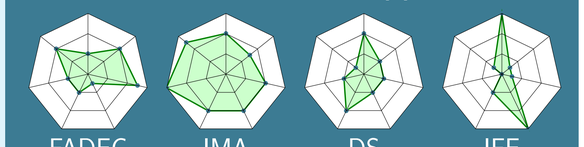


- During a timeslot, only one of the running tasks can access shared hardware resources
- Cache locking & MMU programming

- TDMA time partitioning
- No modification to legacy software
- Limited efficiency



Evaluation vs. Avionic Applications



	Good	Average	Bad
Memguard	IFE	DS	FADEC,IMA
D.R.W.C	DS, IFE	-	FADEC,IMA
D.E.M	FADEC, IMA	DS	IFE
D.A.S	FADEC, IFE	IMA	DS
Marthy	IMA	FADEC, DS	IDE

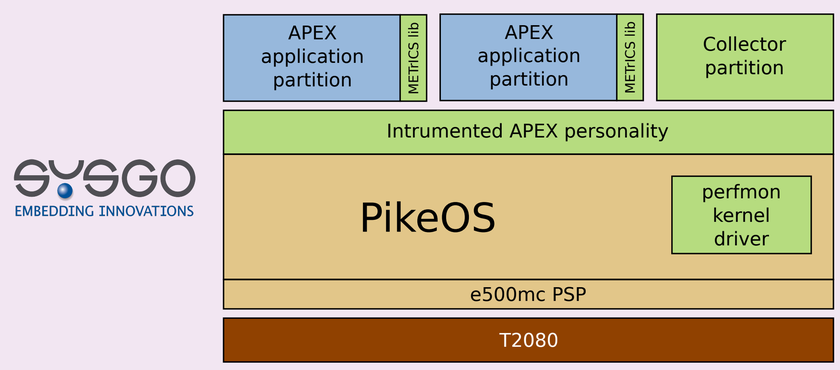
- Survey of Deterministic Platform Software
- All proposed solutions are reaching a different compromise on identified key properties
- With different level of maturity
- Qualitative approach is not sufficient

Application Performance Profiling Solution

METrICS: Measure Environment for Time-Critical Systems

Avionics software specifics

- Safety standards require applications to be partitioned, typically in time domain
- APEX is the API specified by Arinc 653 norm for application partitioning
- Multicore architecture breaks strict time isolation paradigm



SYSGO EMBEDDING INNOVATIONS

NXP

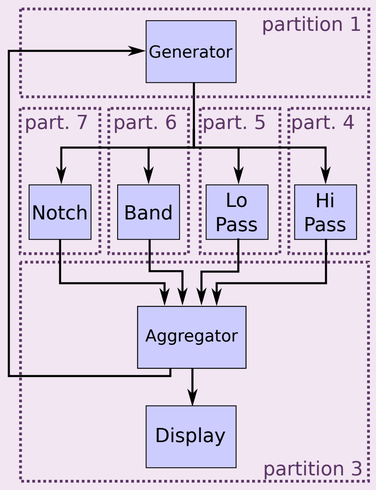
Solution

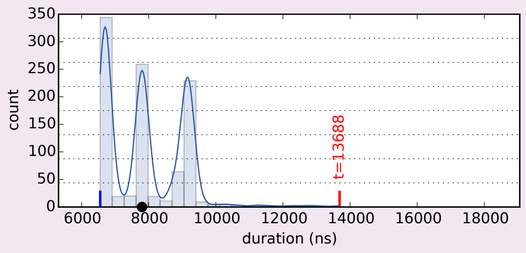
- Use of cycle-accurate core timer and hardware performance counters, for minimal timing overhead. A kernel driver has been developed for configuration of HW counters, and a data collection infrastructure has been set up.
- Automatic instrumentation of all APEX system calls
- Statistical analysis of execution time distribution: full histogram is kept

Preliminary Results on a control-command application

- Experimental application exhibiting all APEX intra- and inter-partition communications, and floating-point calculations.

Inter-partition parallelism



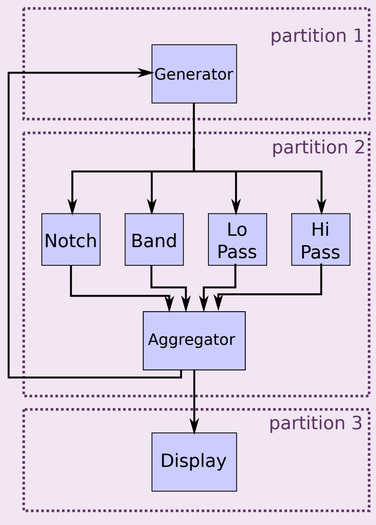


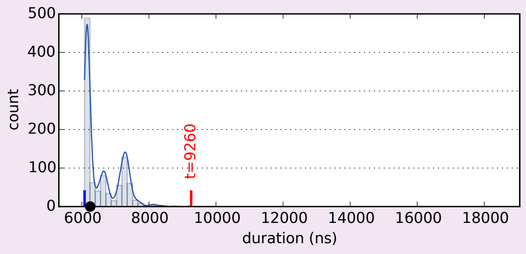
- Experimental application split into several resource partitions
- Mapping-agnostic communication mechanisms
- Processor-space isolation of partitions
- Memory hierarchy is the main shared resource

Results

- Large variability of execution time
- Impact of inter-partition communications on WCET
- Source of variability to be determined by analysis of hardware performance counters

Intra-partition parallelism





- Experimental application with parallel APEX processes within a partition
- Compliant with newest Arinc-653 P1-4 standard, with processor affinity
- Isolation not required within a partition, but same application
- Mapping-dependant communication mechanisms

Results

- Varying amount of interference among cores
- Lower variability than inter-partition parallelism

Conclusions

- Capability to monitor avionics application behaviour with low timing overhead
- Comparison of multicore deployment options
- Timing characterization only allows to quantify interference
- Hardware monitoring is necessary to identify interference sources