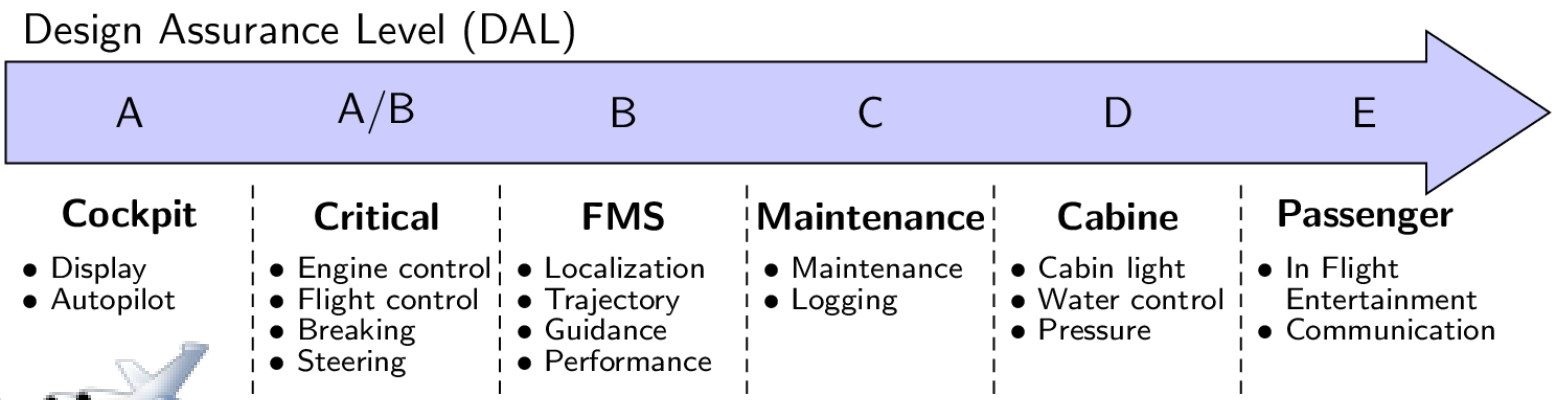
ETH zürich

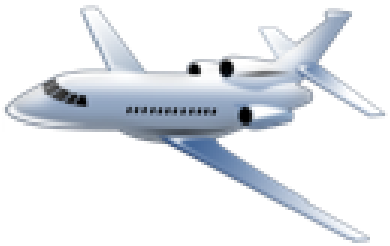
Mixed-Criticality Runtime Mechanisms and Evaluation on Multicores

L. Sigrist, G. Giannopoulou, P. Huang, A. Gomez, L. Thiele
Computer Engineering and Networks Laboratory, ETH Zurich

Mixed-Criticality Systems

- Integration of mixed-criticality (MC) applications into a common platform
- Criticality level (CL) expresses required protection against failure





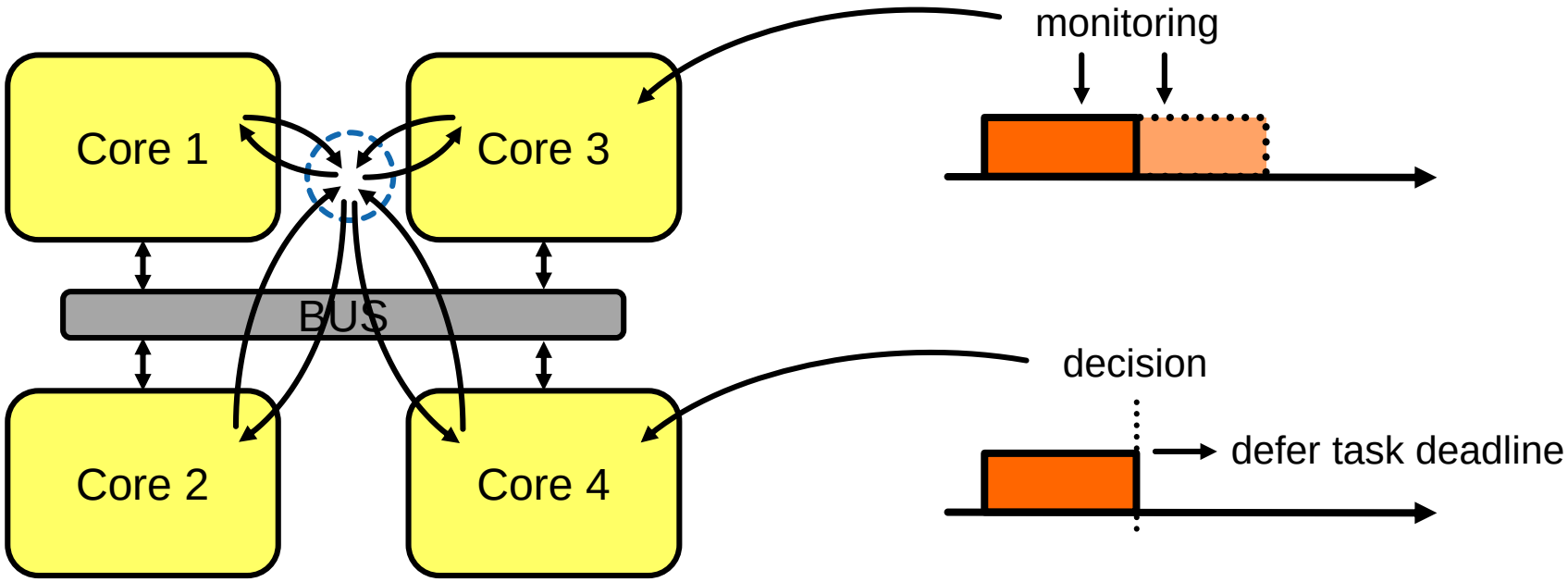
© THALES

Scheduling Mixed-Criticality Tasks

- Challenges
  - Independence between the different CL
  - Resource efficiency

- More complex runtime mechanisms needed
  - Monitoring
  - Mode switching
  - Global synchronization
  - Task termination

Mixed-Criticality Services



Related Work

- Multicore Mixed-Criticality Scheduling Policies
  - R.M. Pathan, *Schedulability Analysis of Mixed-Criticality Systems on Multiprocessors*, ECRTS 2012.
  - G. Giannopoulou et al., *Scheduling of Mixed-Criticality Applications on Resource-Sharing Multicore Systems*, EMSOFT 2013.
  - S. Baruah et al., *Mixed-criticality scheduling on multiprocessors*, RTS 2014.
- Mixed-Criticality Scheduling Frameworks
  - J. Anderson et al., *Multicore operating-system support for mixed criticality,* WMC 2009.
  - J. Herman et al., *RTOS Support for Multicore Mixed-Criticality Systems*, RTAS 2012.
  - H.-M. Huang et al., *Implementation and Evaluation of Mixed-criticality Scheduling Approaches for Sporadic Tasks*, TECS 2014.

Motivation

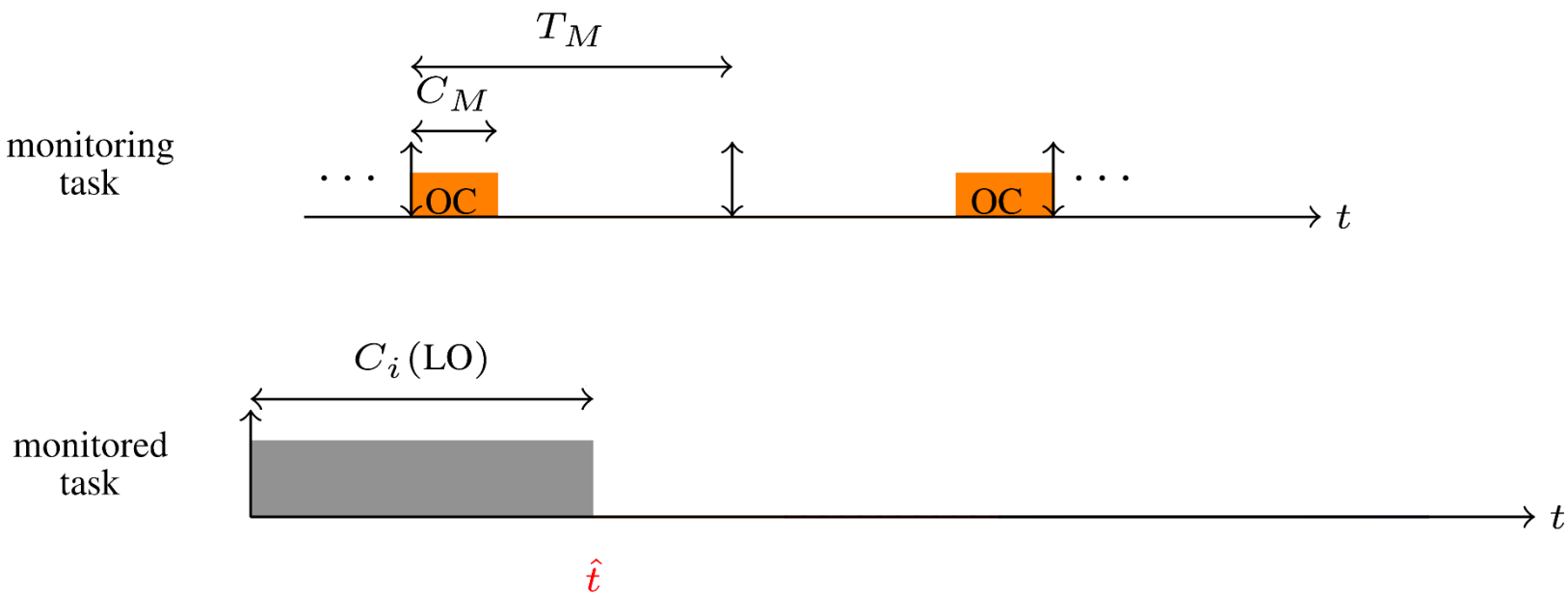
How can we implement mixed-criticality mechanisms **efficiently** on **multicore** systems?

How can we quantify their **runtime overheads** and their **impact on schedulability**?

Motivational Example

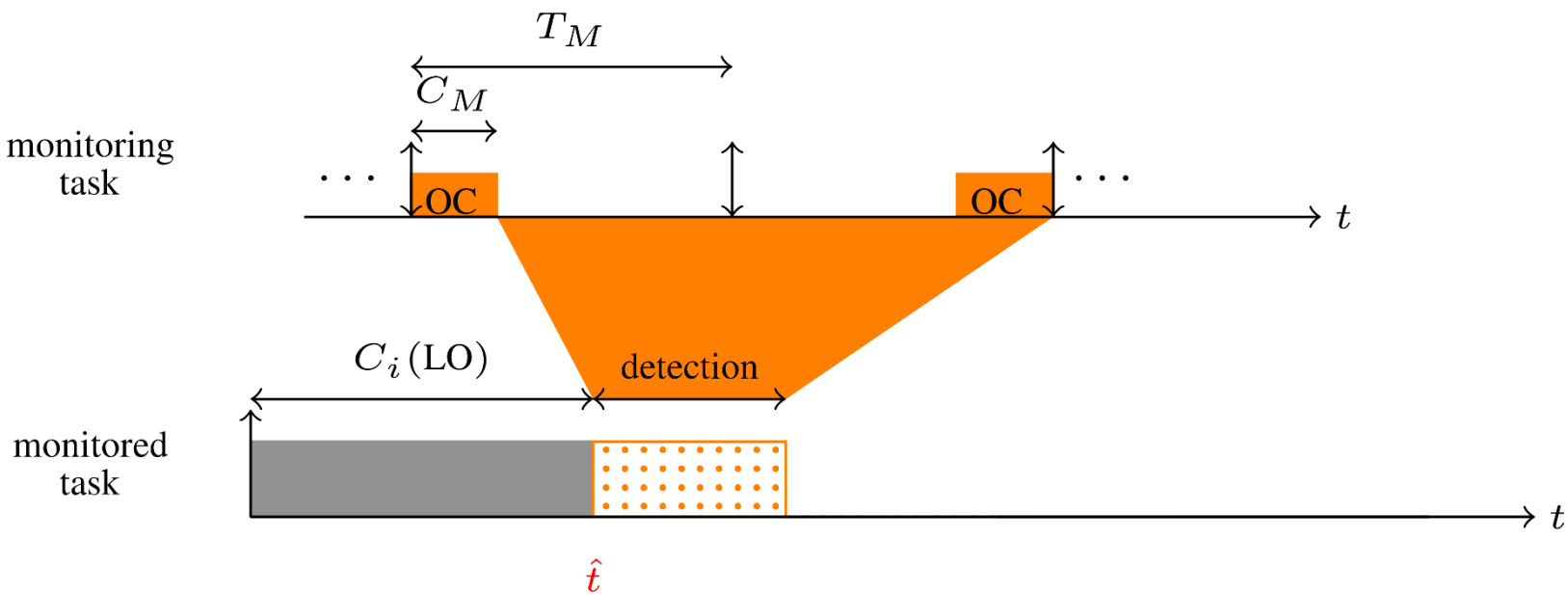
- Monitored task τ_i: $C_i(\mathrm{LO})$, $C_i(\mathrm{HI})$ for LO/HI criticality WCET
  - Overrun at $\hat{t}$, mode switch delayed by detection and termination C_K

Motivational Example



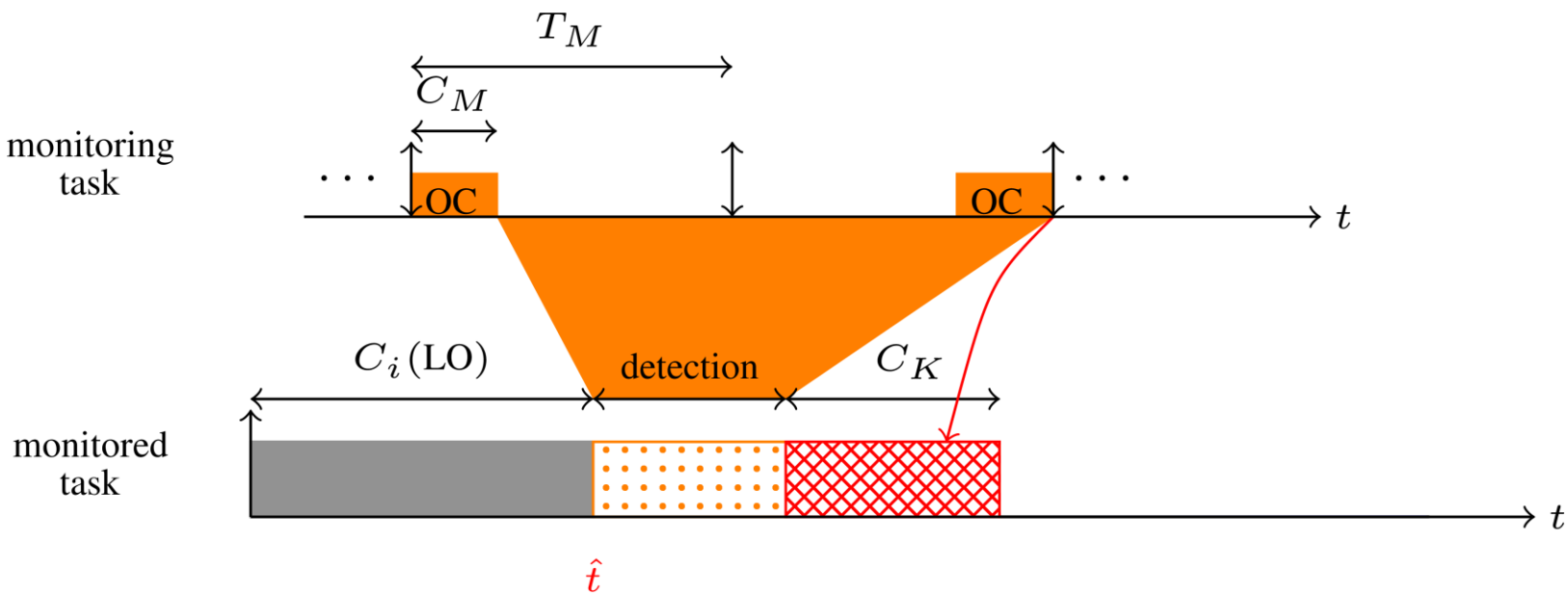
- Monitored task τ_i: $C_i(\mathrm{LO})$, $C_i(\mathrm{HI})$ for LO/HI criticality WCET
  - Overrun at $\hat{t}$, mode switch delayed by detection and termination C_K

Motivational Example



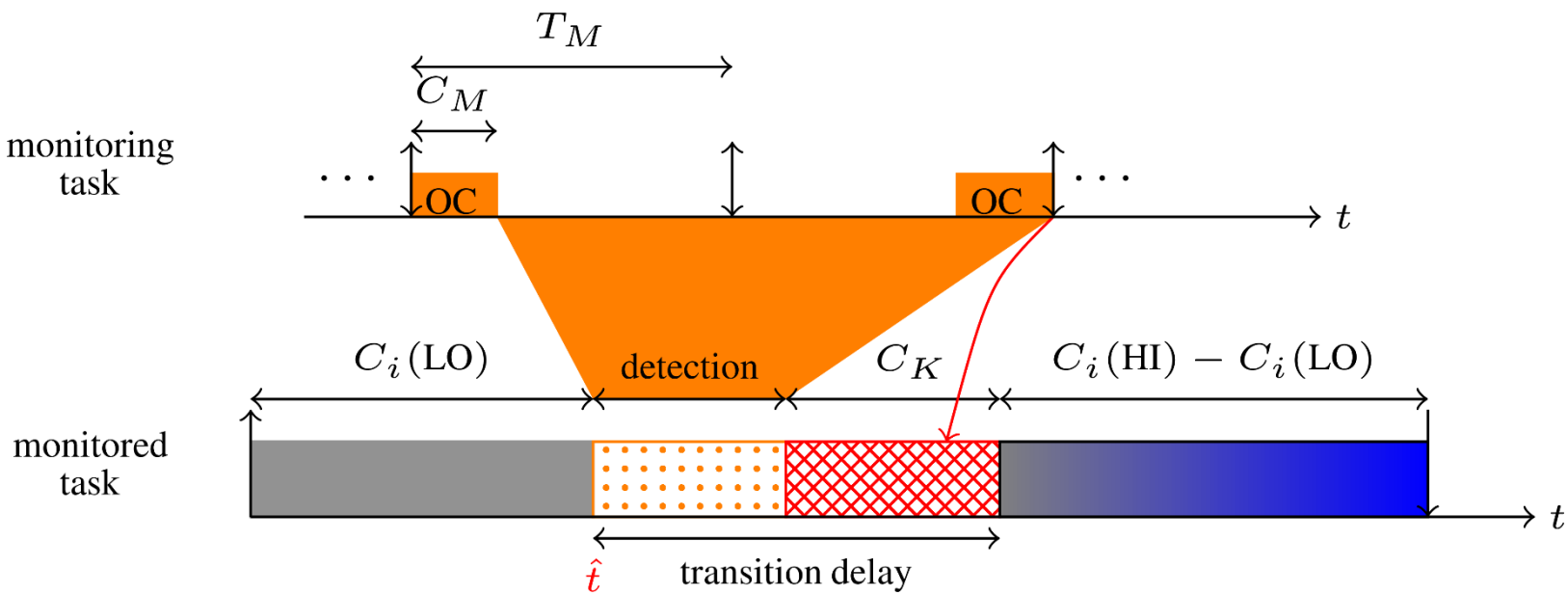
- Monitored task τ_i: $C_i(\mathrm{LO})$, $C_i(\mathrm{HI})$ for LO/HI criticality WCET
  - Overrun at $\hat{t}$, mode switch delayed by detection and termination C_K

Motivational Example



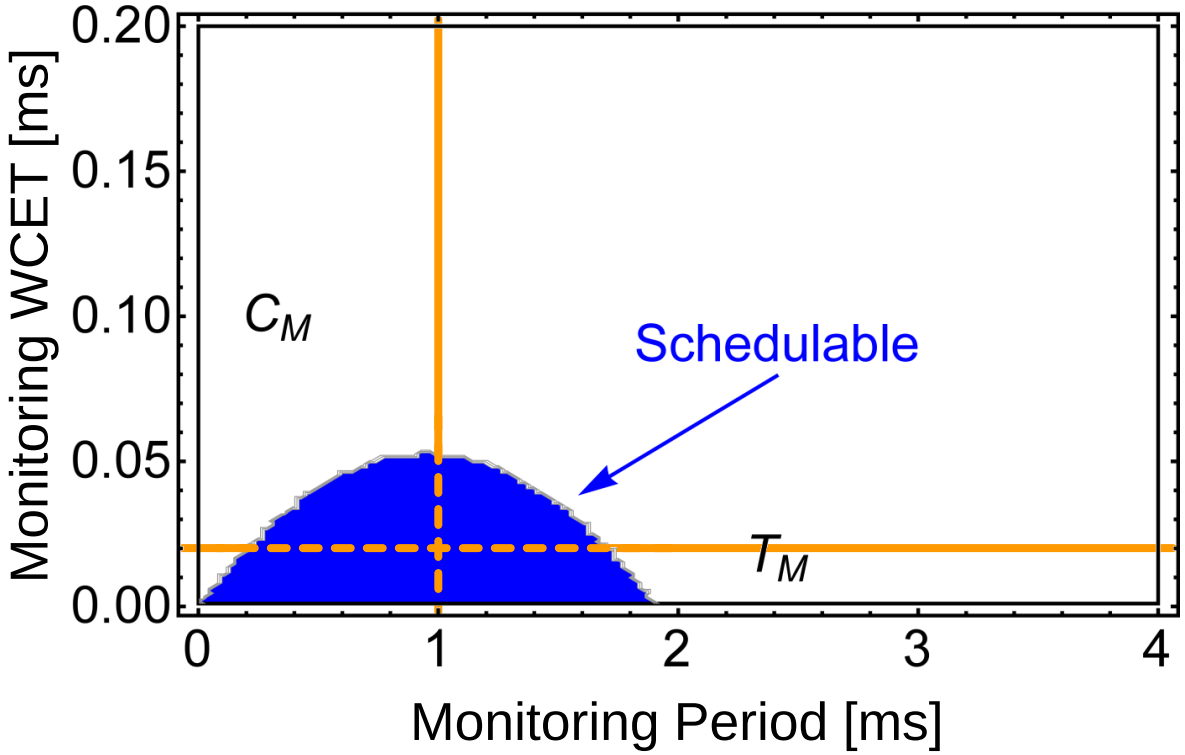
- Monitored task τ_i: $C_i(\mathrm{LO})$, $C_i(\mathrm{HI})$ for LO/HI criticality WCET
  - Overrun at $\hat{t}$, mode switch delayed by detection and termination C_K

Motivational Example



- Monitored task τ_i: $C_i(\mathrm{LO})$, $C_i(\mathrm{HI})$ for LO/HI criticality WCET
  - Overrun at $\hat{t}$, mode switch delayed by detection and termination C_K

Motivational Example: Monitoring Overhead



Termination overhead $C_K = 3\ \mathrm{ms}$

Outline

- Scheduling Policies
  - Flexible Time-Triggered and Synchronization-based (FTTS) scheduler
  - Partitioned EDF with Virtual Deadlines (pEDF-VD)
- Mixed-Criticality Runtime Mechanisms
  - Execution Time Monitoring
  - Scheduler Mode Switch
  - Task Termination
- Mixed-Criticality Scheduling Framework
- Evaluation on Multicore Systems
  - Estimation of Runtime Overheads
  - Effect of Overheads on Schedulability
  - Avionics Case Study

Outline

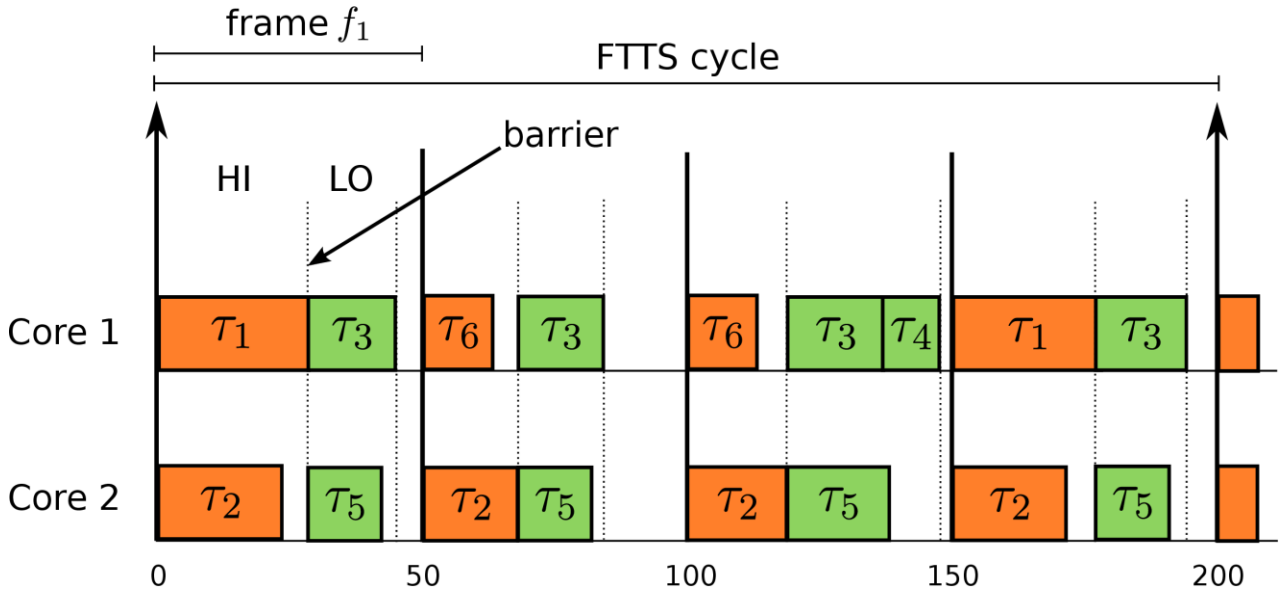
- Scheduling Policies
  - Flexible Time-Triggered and Synchronization-based (FTTS) scheduler
  - Partitioned EDF with Virtual Deadlines (pEDF-VD)
- Mixed-Criticality Runtime Mechanisms
  - Execution Time Monitoring
  - Scheduler Mode Switch
  - Task Termination
- Mixed-Criticality Scheduling Framework
- Evaluation on Multicore Systems
  - Estimation of Runtime Overheads
  - Effect of Overheads on Schedulability
  - Avionics Case Study

Task Model

- Periodic task sets $\tau = \{\tau_1, \dots, \tau_n\}$
- Task $\tau_i = \{T_i, \chi_i, C_i(\mathrm{LO}), C_i(\mathrm{HI})\}$
  - Period T_i
  - Criticality level $\chi_i \in \{\mathrm{LO}, \mathrm{HI}\}$
  - Worst case execution time (WCET) at LO criticality $C_i(\mathrm{LO})$
  - WCET at HI criticality $C_i(\mathrm{HI})$ ($C_i(\mathrm{HI}) = 0$ for $\chi_i = \mathrm{LO}$)
  - Deadline equal to period $D_\mathrm{i} = T_i$

Flexible Time-Triggered FTTS Scheduler [1]

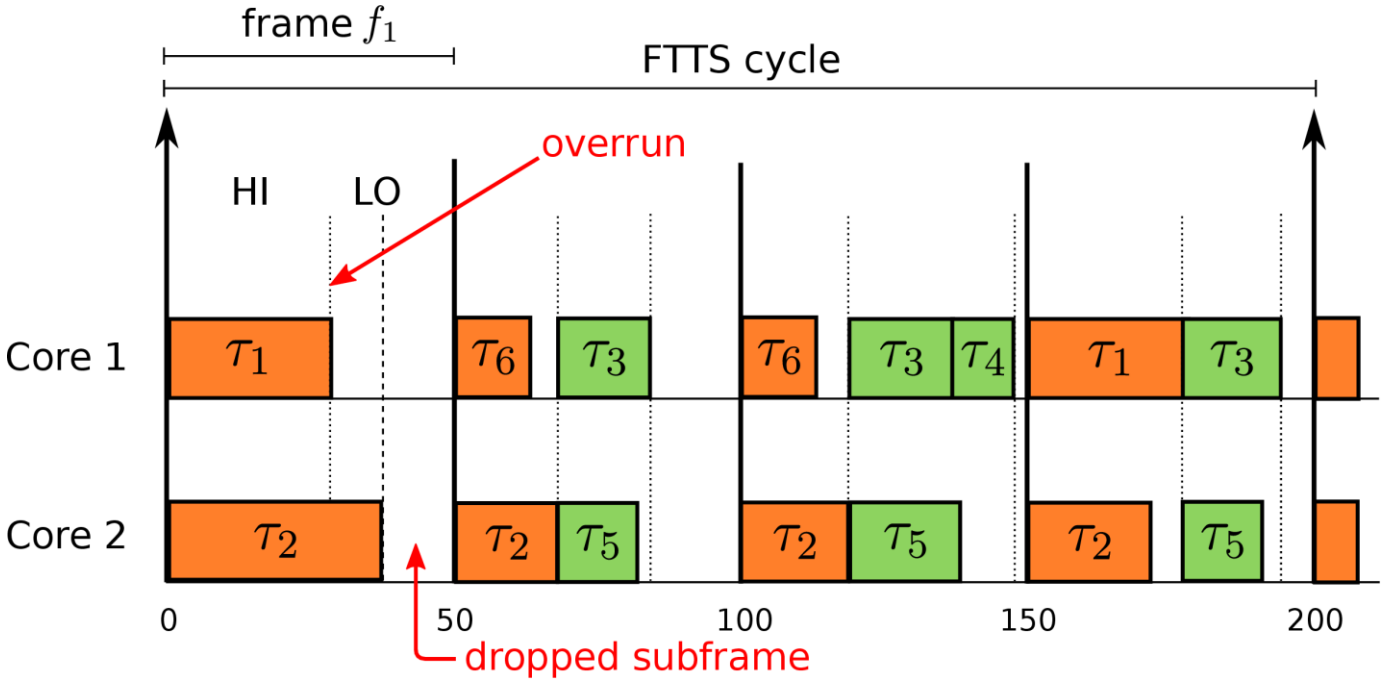
- Sub-frames for criticality levels



[1] G. Giannopoulou, N. Stoimenov, P. Huang, and L. Thiele. *Scheduling of Mixed-Criticality Applications on Resource-Sharing Multicore Systems*. 2013.

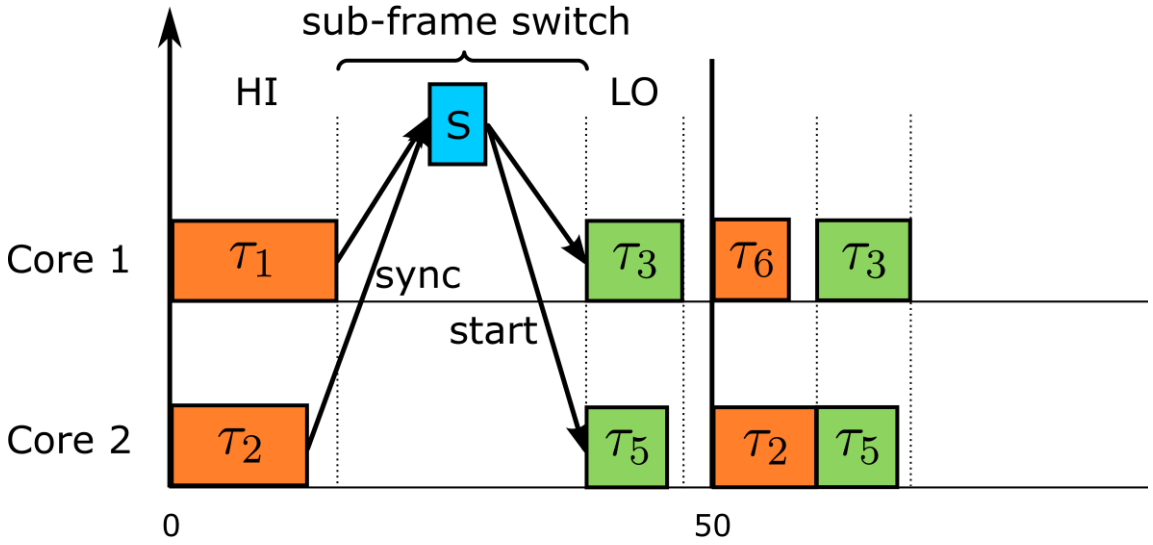
Flexible Time-Triggered FTTS Scheduler

- Sub-frames for criticality levels
- Skip LO sub-frame on HI overrun



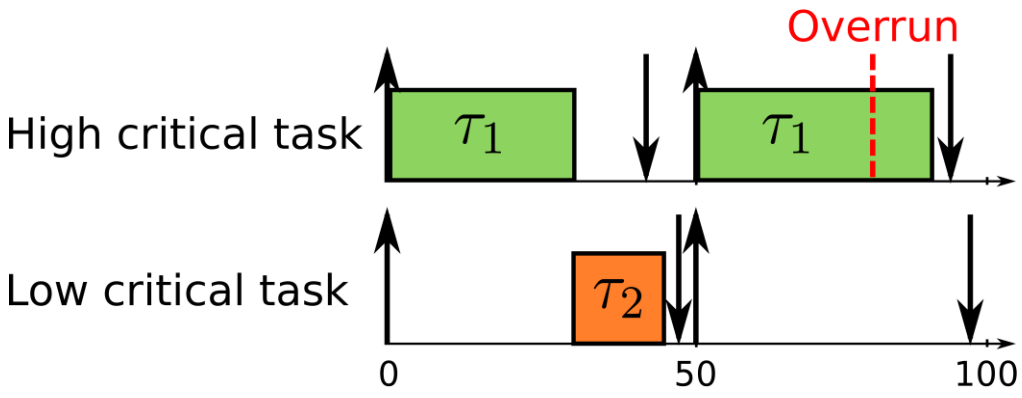
Flexible Time-Triggered FTTS Scheduler

- Sub-frames for criticality levels
- Skip LO sub-frame on HI overrun
- Global decision at sub-frame switch



Partitioned EDF with Virtual Deadlines [2]

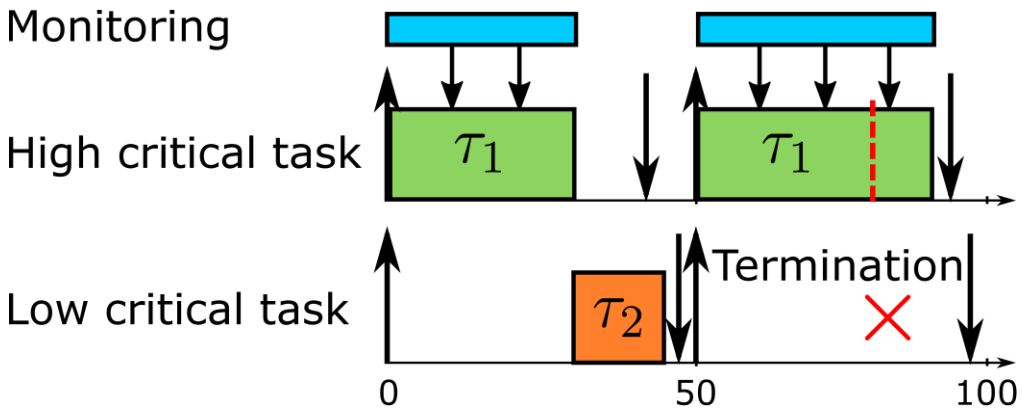
- Based on single core EDF with Virtual Deadlines
  - Mode switch on LO WCET overrun



[2] S. Baruah, B. Chattopadhyay, H. Li, and I. Shin. *Mixed-criticality scheduling on multiprocessors*. 2014.

Partitioned EDF with Virtual Deadlines [2]

- Based on single core EDF with Virtual Deadlines
  - Mode switch on LO WCET overrun



- Offline partitioning
  - First fit bin packing algorithm
  - Independent execution

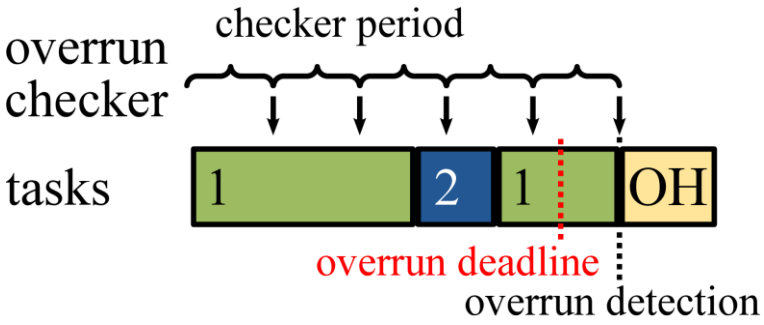
[2] S. Baruah, B. Chattopadhyay, H. Li, and I. Shin. *Mixed-criticality scheduling on multiprocessors*. 2014.

Outline

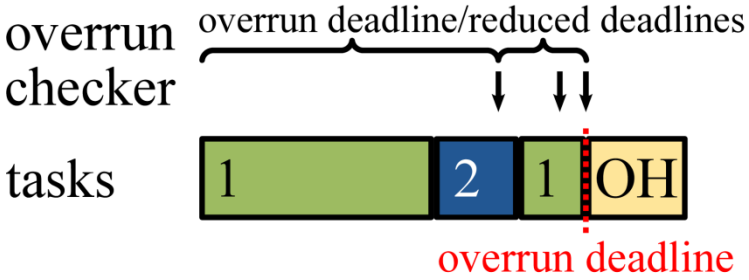
- Scheduling Policies
  - FTTS
  - pEDF-VD
- Mixed-Criticality Runtime Mechanisms
  - Execution Time Monitoring
  - Scheduler Mode Switch
  - Task Termination
- Mixed-Criticality Scheduling Framework
- Evaluation on Multicore Systems
  - Estimation of Runtime Overheads
  - Effect of Overheads on Schedulability
  - Avionics Case Study

Runtime Monitoring

- Heartbeat monitoring
  - Integration in analysis
  - Delayed detection

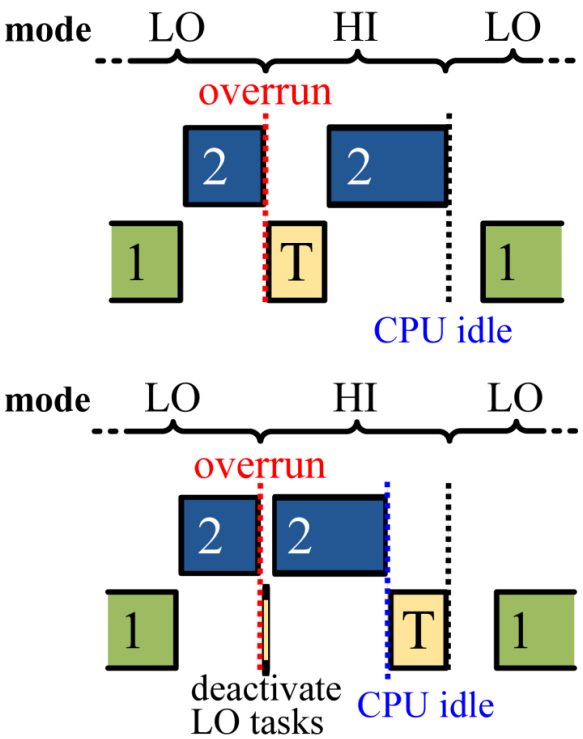


- Deadline-based monitoring
  - Detection at overrun
  - Reduced checks and context switches
  - Hard to integrate in analysis



Task Termination

- Extreme case of degradation
- Immediate Termination
  - No execution before next arrival
  - Additional overhead at time critical mode switch
- Deferred Termination
  - Fast mode switch
  - Tidy up when CPU idle

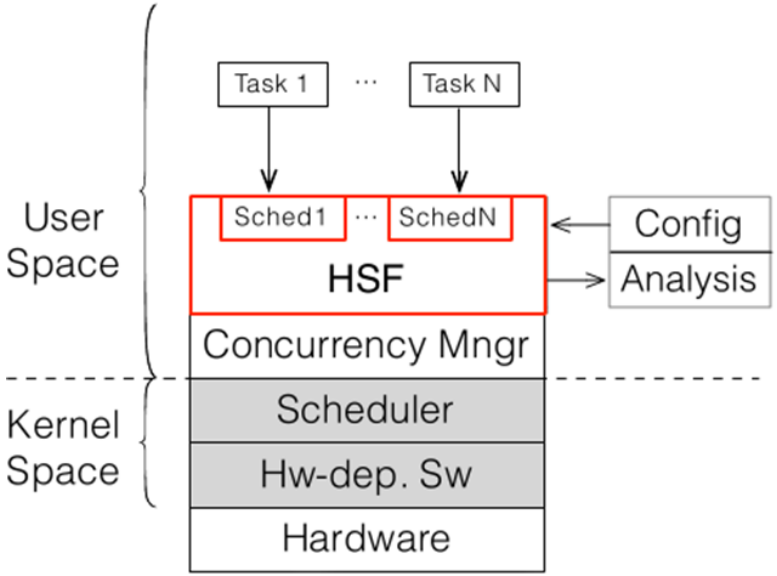


Outline

- Scheduling Policies
  - FTTS
  - pEDF-VD
- Mixed-Criticality Runtime Mechanisms
  - Execution Time Monitoring
  - Scheduler Mode Switch
  - Task Termination
- Mixed-Criticality Scheduling Framework
- Evaluation on Multicore Systems
  - Estimation of Runtime Overheads
  - Effect of Overheads on Schedulability
  - Avionics Case Study

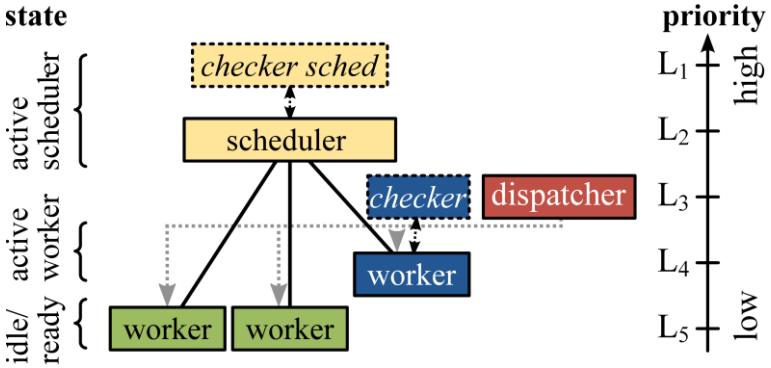
Framework Implementation

- Extension of existing SF3P [3]
- User space framework
  - Portable
  - Easily extensible
  - Rapid prototyping and testing
- Features
  - Mixed-criticality mechanisms
  - Multicore support
  - Execution time tracing



[3] A. Gomez, L. Schor, P. Kumar and L. Thiele. *SF3P: A Framework to Explore and Prototype Hierarchical Compositions of Real-Time Schedulers.* 2014.

Framework Implementation



- Scheduling
  - Priority based kernel scheduler
- Communication
  - Shared memory (mutexes, semaphores)
- Synchronization
  - POSIX barrier (futex based user-space mechanism)

Outline

- Scheduling Policies
  - FTTS
  - pEDF-VD
- Mixed-Criticality Runtime Mechanisms
  - Execution Time Monitoring
  - Scheduler Mode Switch
  - Task Termination
- Mixed-Criticality Scheduling Framework
- Evaluation on Multicore Systems
  - Estimation of Runtime Overheads
  - Effect of Overheads on Schedulability
  - Avionics Case Study

Experiments for Overhead Evaluation

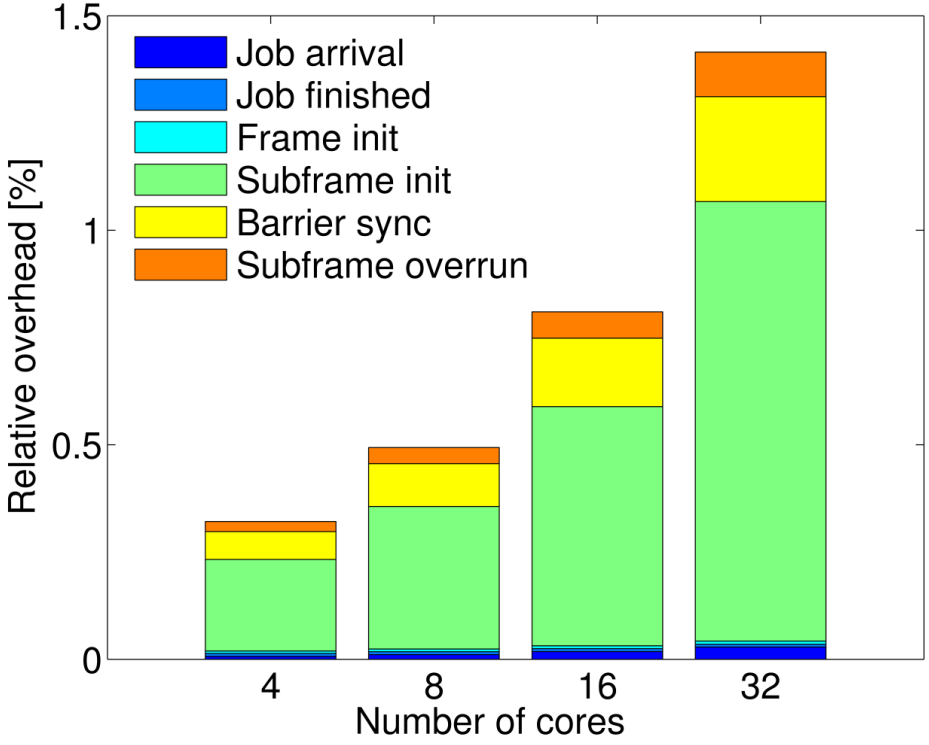
- Simulation
  - 100 task sets per utilization
  - 4-32 cores
  - System utilization from 0.2 to 32.0
  - 40% of tasks with HI criticality level
  - Execution for 10 seconds
- COTS platforms
  - Intel Xeon Phi Coprocessor 5110P (60 cores)
    - Linux 2.6.38.8, GNU libc v2.14
  - Intel Core i5-4670 (4 cores)
    - Linux 3.13.0, GNU libc v2.19

Overhead Scaling with Number of Cores (FTTS)

Average overhead across cores

- Configuration
  - Utilization 0.8-6.4
  - Overrun probability 30%

Global overheads dominate



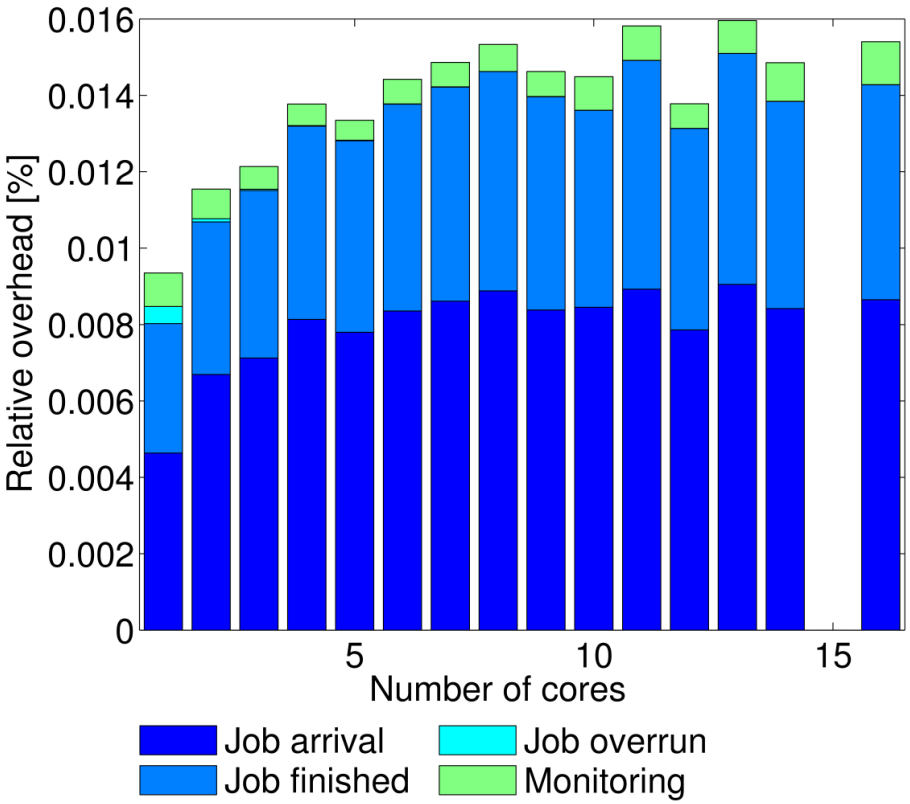
Overhead Scaling with Number of Cores (pEDF-VD)

Average overhead across cores

- Configuration
  - Utilization 0.2-12.0
  - Overrun probability 50%
  - Cores after partitioning

Distributed monitoring keeps overhead low

Deferred termination results in fast mode switch



Impact on pEDF-VD Schedulability

- Negligible impact for pEDF-VD
  - Low overhead, no inter-core communication
  - Conservative schedulability condition for partitioning
- No deadline overruns observed
  - Theoretical schedulable executed

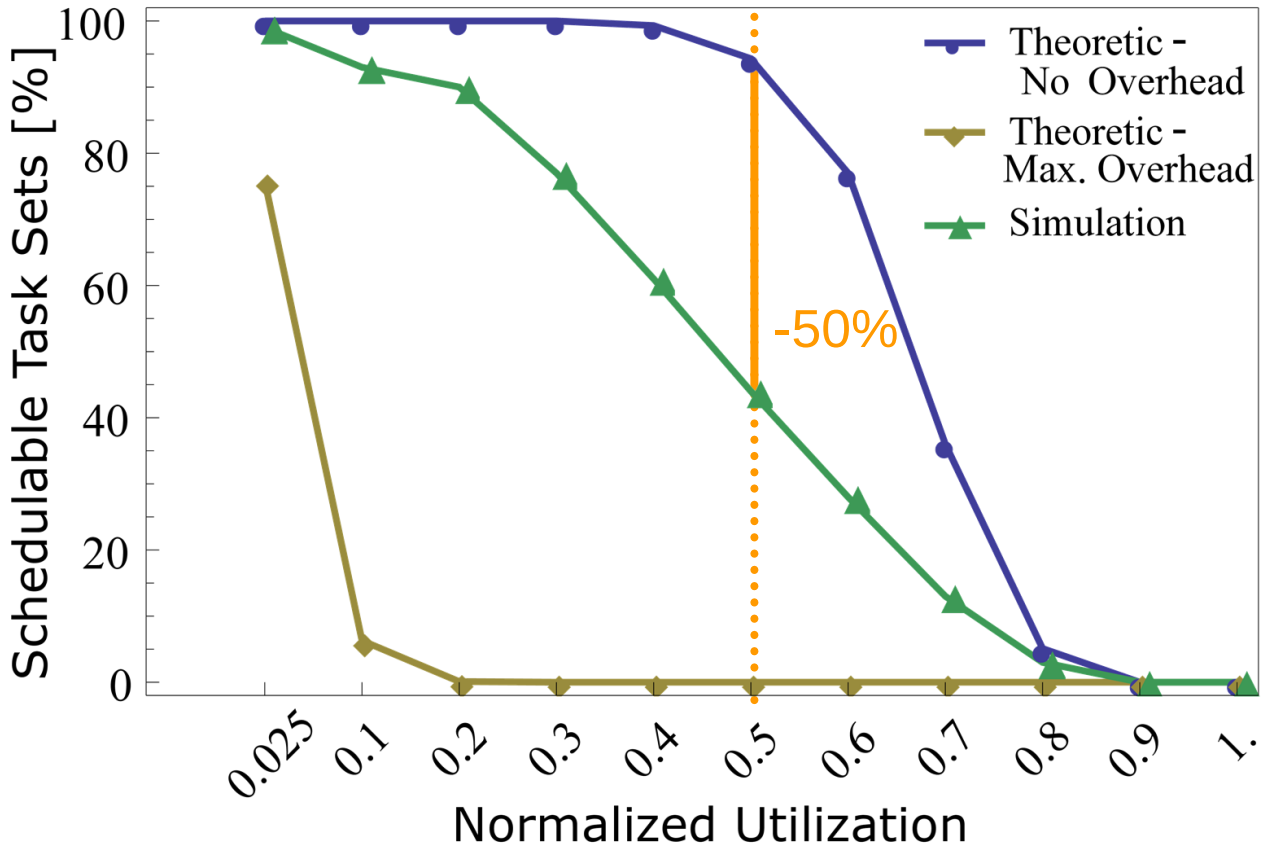
Impact on FTTS Schedulability

- Theoretical schedulability incl. barrier overheads
  - Average overhead
  - Worst case overhead
- Compare to empirical schedulability

Synchronization and sub-frame overhead

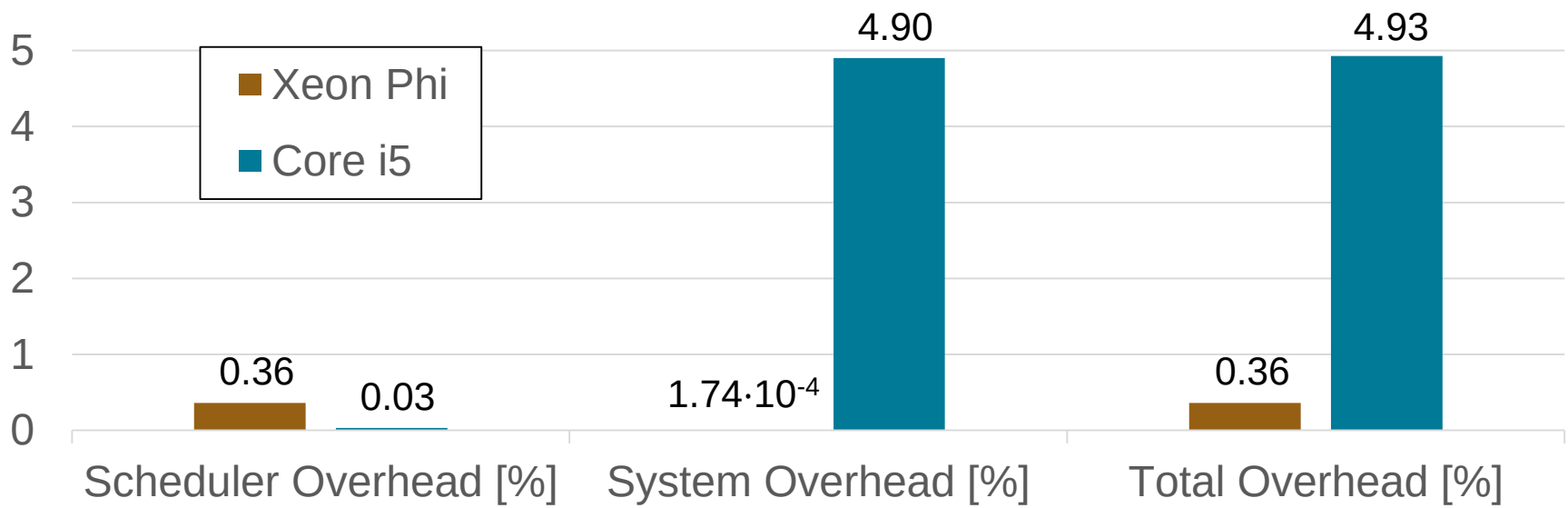
Cores	Average [ms]	Maximum [ms]
4	1.7	45
8	2.5	126
16	3.2	143
32	5	160

Impact on FTTS Schedulability (8 Cores)



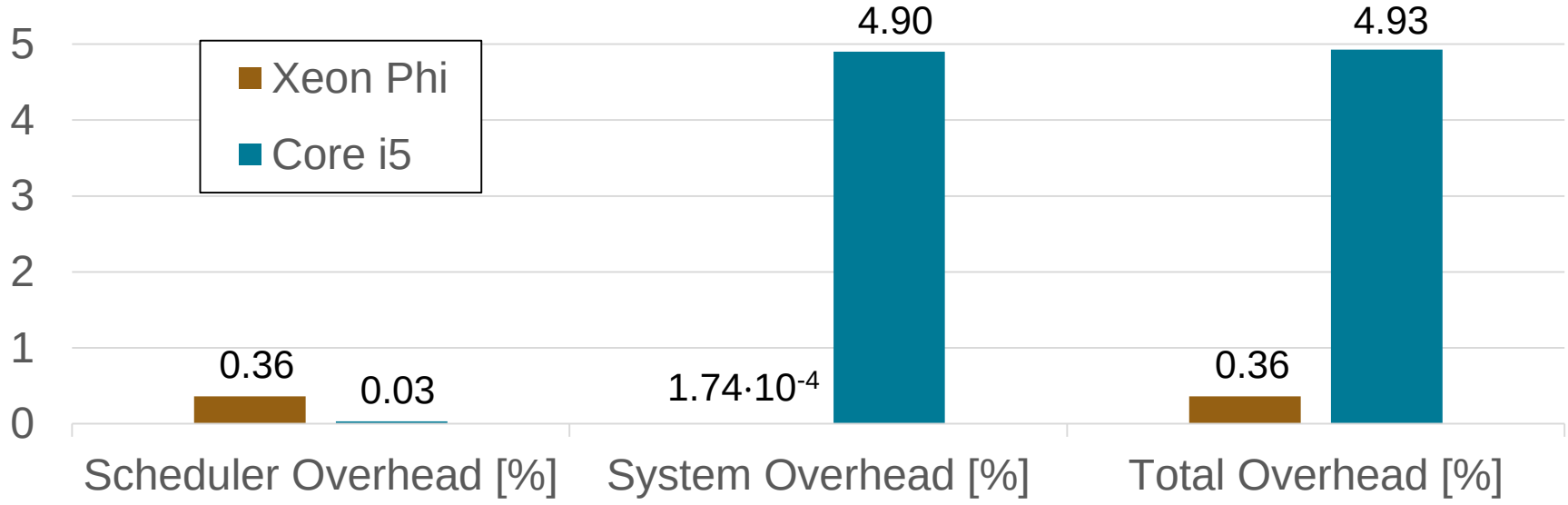
Industrial Avionics Example

- Flight Management System
  - 11 periodic tasks (200 ms, 1 s, 5 s periods)
  - FTTS optimization for 4 cores
  - Extra memory allocation for HI tasks in 50% of executions



Industrial Avionics Example

Rapid prototyping and fast comparison of policies and available platforms



Conclusion

- Implementation alternatives for MC mechanisms
- User-space framework
  - Rapid prototyping and comparison of policies and implementations
  - Fast platform evaluation
- Consider runtime overheads at design time
  - Affect the schedulability
- Efficient, timing-predictable mechanisms needed
  - Operating system
  - Platform