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

### Design Assurance Level (DAL)

- **A**
  - Cockpit
    - Display
    - Autopilot

- **A/B**
  - Critical
    - Engine control
    - Flight control
    - Breaking
    - Steering

- **B**
  - FMS
    - Localization
    - Trajectory
    - Guidance
    - Performance

- **C**
  - Maintenance
    - Maintenance
    - Logging

- **D**
  - Cabine
    - Cabin light
    - Water control
    - Pressure

- **E**
  - Passenger
    - In Flight Entertainment
    - Communication
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

- Core 1
- Core 2
- Core 3
- Core 4

BUS

monitoring

decision

defer task deadline
Related Work

- Multicore Mixed-Criticality Scheduling Policies

- Mixed-Criticality Scheduling Frameworks
  - J. Anderson et al., *Multicore operating-system support for mixed criticality*, WMC 2009.
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 $\tau_i$: $C_i$(LO), $C_i$(HI) for LO/HI criticality WCET
- Overrun at $\hat{t}$, mode switch delayed by detection and termination $C_K$
Motivational Example

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

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

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

- Monitored task $\tau_i$: $C_i(\text{LO})$, $C_i(\text{HI})$ for LO/HI criticality WCET
- Overrun at $\hat{t}$, mode switch delayed by detection and termination $C_K$
Motivational Example: Monitoring Overhead

Termination overhead $C_K = 3$ 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 \( \tau = \{\tau_1, ..., \tau_n\} \)

- Task \( \tau_i = \{T_i, \chi_i, C_i(\text{LO}), C_i(\text{HI})\} \)
  - Period \( T_i \)
  - Criticality level \( \chi_i \in \{\text{LO, HI}\} \)
  - Worst case execution time (WCET) at LO criticality \( C_i(\text{LO}) \)
  - WCET at HI criticality \( C_i(\text{HI}) \) (\( C_i(\text{HI}) = 0 \) for \( \chi_i = \text{LO} \))
  - Deadline equal to period \( D_i = T_i \)
Flexible Time-Triggered FTTS Scheduler \cite{1}

- Sub-frames for criticality levels

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

Partitioned EDF with Virtual Deadlines \cite{2}

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

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

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

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

Lukas Sigrist, Computer Engineering Group, ETH Zurich | 15.04.2015 | 24
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

- 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

<table>
<thead>
<tr>
<th>Cores</th>
<th>Average [ms]</th>
<th>Maximum [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.7</td>
<td>45</td>
</tr>
<tr>
<td>8</td>
<td>2.5</td>
<td>126</td>
</tr>
<tr>
<td>16</td>
<td>3.2</td>
<td>143</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>160</td>
</tr>
</tbody>
</table>

Synchronization and sub-frame overhead
Impact on FTTS Schedulability (8 Cores)

-50%
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

<table>
<thead>
<tr>
<th></th>
<th>Xeon Phi</th>
<th>Core i5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduler Overhead</td>
<td>0.36</td>
<td>0.03</td>
</tr>
<tr>
<td>System Overhead</td>
<td>$1.74 \times 10^{-4}$</td>
<td>$4.90$</td>
</tr>
<tr>
<td>Total Overhead</td>
<td>0.36</td>
<td>4.93</td>
</tr>
</tbody>
</table>
Industrial Avionics Example

Rapid prototyping and fast comparison of policies and available platforms

<table>
<thead>
<tr>
<th></th>
<th>Xeon Phi</th>
<th>Core i5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduler Overhead [%]</td>
<td>0.36</td>
<td>0.03</td>
</tr>
<tr>
<td>System Overhead [%]</td>
<td>1.74\times 10^{-4}</td>
<td>4.90</td>
</tr>
<tr>
<td>Total Overhead [%]</td>
<td>0.36</td>
<td>4.93</td>
</tr>
</tbody>
</table>
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