Partitioned and Overhead-Aware Scheduling of Mixed-Criticality Real-Time Systems

Yuanbin Zhou\textsuperscript{1}, Soheil Samii\textsuperscript{1,2}, Petru Eles\textsuperscript{1}, Zebo Peng\textsuperscript{1}

\textsuperscript{1}ESLAB, Linköping University, Sweden
\textsuperscript{2}General Motors R&D, USA
Overview

1. Introduction & System Model
2. Motivation
3. Problem Formulation
4. Proposed Approach
5. Experimental Results
Mixed-Criticality Systems (MCS)

- Tasks with different criticalities share computation resources
- Criticality used in functional safety, e.g., automotive
- Industrial safety standards
  - ISO 26262 – road vehicles
  - DO 178C – avionics software
  - IEC 61508 – generic standard
- Core concept for MCS is sufficient independence
Partitioned Scheduling (Hierarchical Scheduling)

- **Global Scheduler** assign system resources to Local Scheduler
- Tasks scheduled by **Local Scheduler**
- Misbehaviors do not affect tasks with different criticality levels
- Online local scheduler (flexible), Offline global scheduler (predictable)
Task Model

- Single-core platform
- Timing parameters for task $\tau_i$
  - Worst-case Execution Time (WCET) $C_i$
  - Period $T_i$
  - Relative Deadline $D_i$ ($D_i \leq T_i$)
  - Fixed Priority $P_i$
- Criticality level $SIL_i$
Partition Model

- Periodic partition:
  - Period $T_s$
  - Deadline $D_s$ ($D_s = T_s$)
  - Capacity $C_s$

![Diagram of partition model]
Design of Partitions

- Need to be determined
  - Allocation tasks to partitions
  - Period: $T_s$, Capacity: $C_s$

- ($C_s$, $T_s$) pairs of each partition obtained\textsuperscript{1}
  - Cost function of each partition (e.g., Utilization) defined

- Tasks within partitions are guaranteed schedulable

---
\textsuperscript{1}Almeida L, Pedreiras P. Scheduling within temporal partitions: response-time analysis and server design[C]//Emsoft. 2004: 95-103.
Motivation – Offline Partition Scheduling

- Offline scheduling
  - Schedule table determined before system runs
  - Schedule table size affect: **Synthesis time, Memory usage**

<table>
<thead>
<tr>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partition</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>$S_2$</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$S_3$</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>$S_1$</td>
<td>8</td>
<td>11</td>
</tr>
</tbody>
</table>
Motivation – Hyper-Period

- **Hyper-Period**
  - Schedule is same cross Hyper-Period (HP)
  - Least Common Multiple of periods

- **Very long Hyper-Period**
  - Example 1: HP\{15, 16, 31, 32, 33\} = 163680

- **Harmonic relations**
  - \(\frac{T_{i+1}}{T_i} \in \mathbb{N}^+\)
  - Example 2: HP\{16, 16, 32, 32, 32\} = 32

\[
\begin{array}{cccc}
T_{s1} & s_1 & s_2 & s_3 & s_1 \\
\end{array}
\]

Hyper-Period
Motivation – Period vs Utilization

- Partitions’ utilization increased by setting different periods\(^2\)
  - e.g., harmonic periods
- Research problem: to **trade-off** hyper-period length and system schedulability

\(^2\)Yoon M K et al. Holistic design parameter optimization of multiple periodic resources in hierarchical scheduling[C]//DATE,2013: 1313-1318.
Motivation – Overhead between Partitions

- Construct **offline** schedule table
  - Preemptive EDF to simulate within hyper-period
    - … Utilization $\leq 1 \rightarrow$ schedulable
  - Several partition slices due to preemption
  - Scheduling overhead between partition slices

\[ C_{s_1} \]
\[ s_1 \]
\[ T_{s_1} \]
Motivation – Overhead between Partitions (Cont.)

- Too many partition slices → too much scheduling overhead → impact schedulability
- EDF is not optimal when overhead considered
- Combine partition slices → reduce utilization → improve schedulability
  - Deadline and release constraints not violated
  - Possible due to offline scheduling
Problem Formulation

- Input:
  - Task Parameters (including WCET, Deadline, Period, Criticality Level)

- Output:
  - Offline schedule table for partitions

- Constraints:
  - System is schedulable
  - Reduce partition schedule length
Overall Approach

1. Start
2. Construct \((C_s, T_s)\) search space for each partition
3. Hyper-Period optimization
4. Schedule table construction for partitions
5. Overhead optimization
6. Schedulable?
   - no
6.1. Go back to Hyper-Period optimization
   - yes
   - Stop
Experimental Results (Hyper-Period Value)

(a) Proposed Approach

- Proposed approach is within several hundreds
- Straight-forward approach (NIS): rounded into nearest integer
  - from several hundreds to $10^{37}$

(b) Straight-forward Approach
Experimental Results (System Schedulability)

- HPOA: proposed approach, NIS: straight-forward approach
- Differences in schedulability are smaller than 10%
- Large reduction in hyper-period, small sacrifice on schedulability
Experimental Results (Scalability)

- Synthesis is done within scalable time
Conclusions

- Partitioned and overhead-aware scheduling **framework** for mixed-criticality systems
- Synthesis schedule table for partitions with **reduced schedule length** and **preserved system schedulability**
- **Re-scheduling algorithm** to reduce runtime overhead between partitions