Static CRPD-Aware Real-Time Scheduling
ECRTS-WiP

Guillaume PHAVORIN, Pascal RICHARD
LIAS/Université de Poitiers
{guillaume.phavorin,pascal.richard}@univ-poitiers.fr

Claire MAIZA
Verimag/INP Grenoble
claire.maiza@imag.fr

July 8th, 2015
Actual platforms: CPUs + caches

⇒ Cache-Related Preemption Delays (CRPD)

→ cannot be neglected

→ WCET depends on tasks running concurrently (inter-task interference in the cache)

⇒ WCETs are not independent
Actual platforms: CPUs + caches
⇒ Cache-Related Preemption Delays (CRPD)
→ cannot be neglected

→ WCET depends on tasks running concurrently
 (inter-task interference in the cache)
⇒ WCETs are not independent

- **preemption-aware scheduling**: preemption thresholds, deferred preemption, preemption points...

- **CRPD-aware scheduling** → scheduling decisions using CRPD values
The CRPD-aware scheduling problem is defined as:

\[ \tau_i(C_i, T_i, D_i, s_i) \]

- additional task parameter \( s_i \): worst-case CRPD for \( \tau_i \)
- \( C_i \): WCET without CRPD
\( \tau_i(C_i, T_i, D_i, s_i) \)

- additional task parameter \( \rightarrow s_i \): worst-case CRPD for \( \tau_i \)
- \( C_i \): WCET without CRPD

\( \rightarrow \) RM and EDF **not optimal**
**CRPD-aware scheduling problem**

\[ \tau_i(C_i, T_i, D_i, s_i) \]

- additional task parameter \( s_i \): worst-case CRPD for \( \tau_i \)
- \( C_i \): WCET without CRPD

\[ \rightarrow \text{RM and EDF not optimal} \]

\[ \rightarrow \text{NP-hard in the strong sense} \]
A solution using static scheduling

- static scheduling $\rightarrow$ schedule computed offline
A solution using static scheduling

- **static scheduling** $\rightarrow$ schedule computed offline

- **mathematical problem** $\rightarrow$ Mixed-Integer Linear Program (MILP)
  - **schedule** $\rightarrow$ set of slices delimited by releases and deadlines

### Schedule Computation

<table>
<thead>
<tr>
<th>$\tau_1(1,3,3,0.25)$</th>
<th>$J_1(0,1,3,0.25)$</th>
<th>$J_{1,1} \in [0,3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J_2(3,1,6,0.25)$</td>
<td>$J_{2,2} \in [3,6)$</td>
</tr>
<tr>
<td></td>
<td>$J_3(6,1,9,0.25)$</td>
<td>$J_{3,3} \in [6,9)$</td>
</tr>
<tr>
<td></td>
<td>$J_4(9,1,12,0.25)$</td>
<td>$J_{4,4} \in [9,12)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\tau_2(7,12,12,0.5)$</th>
<th>$J_5(0,7,12,0.5)$</th>
<th>$J_{5,1} \in [0,3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J_6(7,12,12,0.5)$</td>
<td>$J_{6,2} \in [3,6)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$J_{6,3} \in [6,9)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$J_{6,4} \in [9,12)$</td>
</tr>
</tbody>
</table>

\[\tau_2 \rightarrow \tau_1\]
A solution using static scheduling

- static scheduling $\rightarrow$ schedule computed offline

- mathematical problem $\rightarrow$ Mixed-Integer Linear Program (MILP)
  - schedule $\rightarrow$ set of slices delimited by releases and deadlines
  $\rightarrow$ jobs scheduled in slices $\rightarrow$ job-pieces

<table>
<thead>
<tr>
<th>$\tau_1(1, 3, 3, 0.25)$</th>
<th>$J_1(0, 1, 3, 0.25)$</th>
<th>$J_1, 1 \in [0, 3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J_2(3, 1, 6, 0.25)$</td>
<td>$J_2, 2 \in [3, 6)$</td>
</tr>
<tr>
<td></td>
<td>$J_3(6, 1, 9, 0.25)$</td>
<td>$J_3, 3 \in [6, 9)$</td>
</tr>
<tr>
<td></td>
<td>$J_4(9, 1, 12, 0.25)$</td>
<td>$J_4, 4 \in [9, 12)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\tau_2(7, 12, 12, 0.5)$</th>
<th>$J_5(0, 7, 12, 0.5)$</th>
<th>$J_5, 1 \in [0, 3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J_5, 2 \in [3, 6)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$J_5, 3 \in [6, 9)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$J_5, 4 \in [9, 12)$</td>
<td></td>
</tr>
</tbody>
</table>
A solution using static scheduling

- static scheduling $\rightarrow$ schedule computed offline

- mathematical problem $\rightarrow$ Mixed-Integer Linear Program (MILP)
  - schedule $\rightarrow$ set of slices delimited by releases and deadlines
    $\rightarrow$ jobs scheduled in slices $\rightarrow$ job-pieces

<table>
<thead>
<tr>
<th>$\tau_1(1, 3, 3, 0.25)$</th>
<th>$J_1(0, 1, 3, 0.25)$</th>
<th>$J_1, 1 \in [0, 3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J_2(3, 1, 6, 0.25)$</td>
<td>$J_2, 2 \in [3, 6)$</td>
</tr>
<tr>
<td></td>
<td>$J_3(6, 1, 9, 0.25)$</td>
<td>$J_3, 3 \in [6, 9)$</td>
</tr>
<tr>
<td></td>
<td>$J_4(9, 1, 12, 0.25)$</td>
<td>$J_4, 4 \in [9, 12)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\tau_2(7, 12, 12, 0.5)$</th>
<th>$J_5(0, 7, 12, 0.5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J_5, 1 \in [0, 3) $</td>
</tr>
<tr>
<td></td>
<td>$J_5, 2 \in [3, 6) $</td>
</tr>
<tr>
<td></td>
<td>$J_5, 3 \in [6, 9) $</td>
</tr>
<tr>
<td></td>
<td>$J_5, 4 \in [9, 12) $</td>
</tr>
</tbody>
</table>

- objective function $\rightarrow$ minimize the overall CRPD
- under several constraints:
  - every job-piece executed in its slice
  - only one job-piece executed at a time
  - CRPD when another job-piece scheduled between 2 job-pieces of a same job

$\rightarrow$ full MILP in the paper
Future works

- evaluate on-line scheduling policies (RM, EDF...)
  - loss of schedulability
  - number of preemptions

- extend the mathematical problem to take into account improved cache analysis (preempted + preempting tasks)

- devise online heuristics scheduling algorithms
Future works

- evaluate on-line scheduling policies (RM, EDF…)
  - loss of schedulability
  - number of preemptions

- extend the mathematical problem to take into account improved cache analysis (preempted + preempting tasks)

- devise online heuristics scheduling algorithms

More information around the poster