Graceful Degradation in Semi-Clairvoyant Scheduling

Sanjoy Baruah
Washington University in Saint Louis

Pontus Ekberg
Uppsala University

ECRTS 2021
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant: (ordinary MC)

Clairvoyant: hi-crit. behavior is signaled here

Semi-clairvoyant: (Agrawal et al., RTSS'19) hi-crit. behavior is signaled here
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant: (ordinary MC)

Clairvoyant:

Semi-clairvoyant: (Agrawal et al., RTSS’19)
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant:
(ordinary MC)

Clairvoyant:

Semi-clairvoyant:
(Agrawal et al., RTSS'19)
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant: (ordinary MC)

Hi-crit. behavior is signaled here

Semi-clairvoyant: (Agrawal et al., RTSS’19)

Hi-crit. behavior is signaled here
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant: (ordinary MC)

Clairvoyant:

HI-crit. behavior is signaled here

HI-crit. behavior is signaled here
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant: (ordinary MC)

Clairvoyant:
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant: (ordinary MC)

Semi-clairvoyant: (Agrawal et al., RTSS’19)

Clairvoyant:
Semi-clairvoyant mixed-criticality scheduling

Non-clairvoyant: (ordinary MC)

Semi-clairvoyant: (Agrawal et al., RTSS’19)

Clairvoyant:
Graceful degradation – what we mean

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with reduced WCET.
Graceful degradation – what we mean

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with reduced WCET.

\[ C(\text{lo}) \leq C(\text{hi}) \]

Lo-criticality:

\[ C(\text{hi}) \leq C(\text{lo}) \]
**Graceful degradation – what we mean**

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with *reduced* WCET.

**Diagram**

**Hi-criticality:**
\[ C(\text{Lo}) \leq C(\text{Hi}) \]

**Lo-criticality:**
\[ C(\text{Hi}) \leq C(\text{Lo}) \]
**Graceful degradation – what we mean**

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with *reduced* WCET.

**Hi-criticality:**
\[ C(\text{Lo}) \leq C(\text{Hi}) \]

**Lo-criticality:**
\[ C(\text{Hi}) \leq C(\text{Lo}) \]
**Graceful degradation – what we mean**

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with *reduced* WCET.

\[
C(\text{Lo}) \leq C(\text{Hi}) \\
C(\text{Hi}) \leq C(\text{Lo})
\]

![Diagram showing graceful degradation](image-url)
Graceful degradation – what we mean

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with reduced WCET.

**Hi-criticality:**
\[ C(\text{Lo}) \leq C(\text{Hi}) \]

**Lo-criticality:**
\[ C(\text{Hi}) \leq C(\text{Lo}) \]
Graceful degradation – what we mean

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with reduced WCET.

Hi-criticality: $C(\text{lo}) \leq C(\text{hi})$

Lo-criticality: $C(\text{hi}) \leq C(\text{lo})$

Smaller WCETs

Larger WCETs
**Graceful Degradation – What We Mean**

Lo-criticality jobs are allowed to execute also after hi-criticality behavior is signaled, but with reduced WCET.

---

**Diagram:**

- **Hi-criticality:** $C(\text{Lo}) \leq C(\text{Hi})$
  - Smaller WCETs
  - Larger WCETs

- **Lo-criticality:** $C(\text{Hi}) \leq C(\text{Lo})$
  - Larger WCETs
  - Smaller WCETs

---

3
Three different correctness criteria

**CC-1**

*No* active lo-crit. job gets to keep their $C_{(lo)}$ budgets.
THREE DIFFERENT CORRECTNESS CRITERIA

**CC-1**

*No* active LO-crit. job gets to keep their $C(\text{LO})$ budgets.

CC-1 reduces to the *standard mixed-criticality semantics* in the absence of graceful degradation (i.e., if $C(\text{HI}) = 0$).
Three different correctness criteria

CC-1

No active Lo-crit. job gets to keep their $C(\text{Lo})$ budgets.

CC-2

Active Lo-crit. jobs that have started execution get to keep their $C(\text{Lo})$ budgets.

CC-3

$CC-1 \Rightarrow CC-2 \Rightarrow CC-3$

CC-1 reduces to the standard mixed-criticality semantics in the absence of graceful degradation (i.e., if $C(\text{Hi}) = 0$).
Three different correctness criteria

**CC-1**

*No* active low-crit. job gets to keep their $C(\text{lo})$ budgets.

**CC-2**

Active low-crit. jobs that have *started execution* get to keep their $C(\text{lo})$ budgets.

**CC-3**

*All* active low-crit. jobs get to keep their $C(\text{lo})$ budgets.
**Three different correctness criteria**

**CC-1**

*No* active low-crit. job gets to keep their $C(\text{lo})$ budgets.

**CC-2**

Active low-crit. jobs that have *started execution* get to keep their $C(\text{lo})$ budgets.

**CC-3**

*All* active low-crit. jobs get to keep their $C(\text{lo})$ budgets.

$$\text{CC-3} \implies \text{CC-2} \implies \text{CC-1}$$
All the problems

We consider *optimal scheduling* and *exact analysis* in different settings:

- With one of three correctness criteria:
  1. CC-1
  2. CC-2
  3. CC-3

- With one of two workload models:
  1. Independent jobs
  2. Sporadic tasks
All the problems

We consider *optimal scheduling* and *exact analysis* in different settings:

- With one of three correctness criteria:
  1. CC-1
  2. CC-2
  3. CC-3

- With one of two workload models:
  1. Independent jobs
  2. Sporadic tasks

- Semi-clairvoyant scheduling

- Graceful degradation

- A preemptive uniprocessor
Correctness criterion CC-1 — insights

No active lo-crit. job gets to keep their $C(\text{lo})$ budgets.
**Correctness criterion CC-1 — insights**

**CC-1**

*No* active *lo-crit.* job gets to keep their $C(\text{lo})$ budgets.

**Lo-crit. job:**
No active low-crit. job gets to keep their $C(\text{lo})$ budgets.
Correctness criterion CC-1 — insights

No active lo-crit. job gets to keep their \( C(\text{lo}) \) budgets.

Hi-crit. job arrives and signals hi-crit. behavior

Lo-crit. job:

\[ C(\text{hi}) \]

\[ C(\text{lo}) \]
Correctness criterion CC-1 — insights

No active lo-crit. job gets to keep their $C(\text{lo})$ budgets.

Hi-crit. job arrives and signals hi-crit. behavior

Can be skipped

lo-crit. job:
Correctness criterion CC-1 — insights

No active lo-crit. job gets to keep their $C(\text{lo})$ budgets.

**CC-1**

Hi-crit. job arrives and signals hi-crit. behavior

Wasted execution

Can be skipped

Lo-crit. job:
CC-1 — RESULTS

Prior work

CC-1 without graceful degradation (i.e., $C(\text{Hi}) = 0$ for low-crit. jobs) is the setting in Agrawal et al., RTSS’19.
CC-1 — RESULTS

Prior work

CC-1 without graceful degradation (i.e., $C(\text{hi}) = 0$ for lo-crit. jobs) is the setting in Agrawal et al., RTSS’19.

We extend the same results to work \textit{with graceful degradation}.
CC-1 — RESULTS

Prior work

CC-1 without graceful degradation (i.e., $C(\text{hi}) = 0$ for low-crit. jobs) is the setting in Agrawal et al., RTSS’19.

We extend the same results to work with graceful degradation.

1. A polynomial-time solution for jobs:
   - A Linear Program (LP) for exactly solving the feasibility problem.
   - A table-based optimal scheduler extracted from the LP solution.
CC-1 — RESULTS

Prior work

CC-1 without graceful degradation (i.e., $C(hi) = 0$ for low-crit. jobs) is the setting in Agrawal et al., RTSS’19.

We extend the same results to work with graceful degradation.

1. A polynomial-time solution for jobs:
   - A Linear Program (LP) for exactly solving the feasibility problem.
   - A table-based optimal scheduler extracted from the LP solution.

2. A polynomial-time solution for implicit-deadline sporadic tasks.
   - An exact utilization-based feasibility test.
   - A fluid-based optimal scheduler.
   - (This followed directly from Agrawal et al.)
Correctness criterion CC-2 — insights

Active lo-crit. jobs that have *started execution* get to keep their $C(\text{lo})$ budgets.
Correctness criterion CC-2 — insights

Active lo-crit. jobs that have \textit{started execution} get to keep their $C(\text{lo})$ budgets.

\begin{itemize}
  \item CC-1
  \begin{itemize}
    \item Wasted execution
    \item Can be skipped
  \end{itemize}
\end{itemize}
Correctness criterion CC-2 — insights

Active lo-crit. jobs that have *started execution* get to keep their $C(\text{lo})$ budgets.

**CC-1**

- Wasted execution
- Can be skipped

**CC-2**

- Can be skipped
Correctness criterion CC-2 — insights

Active lo-crit. jobs that have started execution get to keep their $C(\text{lo})$ budgets.

CC-1

Wasted execution  Can be skipped

CC-2

$C(\text{hi})$  $C(\text{lo})$
Correctness criterion CC-2 — insights

Active lo-crit. jobs that have started execution get to keep their $C(\text{lo})$ budgets.

**CC-1**

Wasted execution  Can be skipped

**CC-2**

Committed execution

$C(\text{hi})$  $C(\text{lo})$  $C(\text{hi})$  $C(\text{lo})$
CC-2 — RESULTS

1. Feasibility for jobs under CC-2 can be solved exactly with an *Mixed Integer Linear Program* (MILP).
CC-2 — RESULTS

1. Feasibility for jobs under CC-2 can be solved exactly with an *Mixed Integer Linear Program* (MILP).

2. A *table-based optimal scheduler* can again be extracted.
CC-2 — RESULTS

1. Feasibility for jobs under CC-2 can be solved exactly with a \textit{Mixed Integer Linear Program} (MILP).

2. A \textit{table-based optimal scheduler} can again be extracted.

3. Feasibility for jobs under CC-2 is \textit{strongly NP-complete}.
CC-2 — RESULTS

1. Feasibility for jobs under CC-2 can be solved exactly with an *Mixed Integer Linear Program* (MILP).

2. A *table-based optimal scheduler* can again be extracted.

3. Feasibility for jobs under CC-2 is *strongly NP-complete*.
Correctness criterion CC-3 — insights

All active lo-crit. jobs get to keep their $C(\text{lo})$ budgets.
Correctness criterion CC-3 — insights

All active lo-crit. jobs get to keep their $C(\text{lo})$ budgets.

**CC-3**

**Observation**

$\Rightarrow$ EDF is an optimal scheduler for both jobs and tasks!
Correctness criterion CC-3 — insights

All active lo-crit. jobs get to keep their $C(\text{lo})$ budgets.

Observation $\Rightarrow$ EDF is an optimal scheduler for both jobs and tasks!
All active lo-crit. jobs get to keep their $C(\text{lo})$ budgets.
Correctness criterion CC-3 — insights

All active lo-crit. jobs get to keep their $C(\text{lo})$ budgets.
Correctness criterion CC-3 — insights

All active lo-crit. jobs get to keep their \( C(\text{lo}) \) budgets.

**Observation**

Under CC-3, the amount of execution time spent on each job is *independent* on the scheduling decisions.
Correctness criterion CC-3 — insights

All active lo-crit. jobs get to keep their $C(\text{lo})$ budgets.

Observation

Under CC-3, the amount of execution time spent on each job is independent on the scheduling decisions.

$\Longrightarrow$ EDF is an optimal scheduler for both jobs and tasks!
Feasibility for *jobs* can be done in $O(n^2 \log n)$ time.
CC-3 — RESULTS FOR JOBS

Jobs

Feasibility for jobs can be done in $O(n^2 \log n)$ time.

(Simply simulate EDF $O(n)$ times.)
Feasibility for *jobs* can be done in $O(n^2 \log n)$ time.

(Simply simulate EDF $O(n)$ times.)
Feasibility for *jobs* can be done in $O(n^2 \log n)$ time.

(Simply simulate EDF $O(n)$ times.)
Feasibility for *jobs* can be done in $O(n^2 \log n)$ time.

(Simply simulate EDF $O(n)$ times.)
Feasibility for *jobs* can be done in $O(n^2 \log n)$ time.

(Simply simulate EDF $O(n)$ times.)
CC-3 — RESULTS FOR JOBS

Feasibility for jobs can be done in $O(n^2 \log n)$ time.

(Simply simulate EDF $O(n)$ times.)
Feasibility for *arbitrary-deadline sporadic tasks* can be done in pseudo-polynomial time if $U$ is bounded by $c < 1$. 
Feasibility for arbitrary-deadline sporadic tasks can be done in pseudo-polynomial time if $U$ is bounded by $c < 1$.

(Based on dbf analysis.)
Which correctness criteria is the correct one?
Which correctness criteria is the correct one?

Jobs

Sporadic tasks

<table>
<thead>
<tr>
<th></th>
<th>CC-1</th>
<th></th>
<th>CC-2</th>
<th></th>
<th>CC-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sporadic tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Which correctness criteria is the correct one?

<table>
<thead>
<tr>
<th></th>
<th>CC-1</th>
<th>CC-2</th>
<th>CC-3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jobs</strong></td>
<td>Polynomial-time solvable (LP formulation)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Table-based optimal scheduler</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sporadic tasks</strong></td>
<td>Simple utilization test</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Optimal fluid scheduler</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Only implicit deadlines)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Which correctness criteria is the correct one?

<table>
<thead>
<tr>
<th></th>
<th>CC-1</th>
<th>CC-2</th>
<th>CC-3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jobs</strong></td>
<td>Polynomial-time solvable (LP formulation)</td>
<td>Strongly NP-complete (MILP formulation)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Table-based optimal scheduler</td>
<td>Table-based optimal scheduler</td>
<td></td>
</tr>
<tr>
<td><strong>Sporadic</strong></td>
<td>Simple utilization test</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>tasks</strong></td>
<td>Optimal fluid scheduler</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Only implicit deadlines)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Which correctness criteria is the correct one?

<table>
<thead>
<tr>
<th></th>
<th>CC-1</th>
<th>CC-2</th>
<th>CC-3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jobs</strong></td>
<td>Polynomial-time solvable (LP formulation)</td>
<td>Strongly NP-complete (MILP formulation)</td>
<td>Solvable in $O(n^2 \log n)$</td>
</tr>
<tr>
<td></td>
<td>Table-based optimal scheduler</td>
<td>Table-based optimal scheduler</td>
<td>EDF optimal</td>
</tr>
<tr>
<td><strong>Sporadic tasks</strong></td>
<td>Simple utilization test</td>
<td></td>
<td>Pseudo-poly. time solvable with bounded utilization</td>
</tr>
<tr>
<td></td>
<td>Optimal fluid scheduler</td>
<td></td>
<td>EDF optimal</td>
</tr>
<tr>
<td></td>
<td>(Only implicit deadlines)</td>
<td></td>
<td>(Even arbitrary deadlines)</td>
</tr>
</tbody>
</table>

$CC-3 \Rightarrow CC-2 \Rightarrow CC-1$
Which correctness criteria is the correct one?

<table>
<thead>
<tr>
<th></th>
<th>CC-1</th>
<th>CC-2</th>
<th>CC-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobs</td>
<td>Polynomial-time solvable (LP formulation)</td>
<td>Strongly NP-complete (MILP formulation)</td>
<td>Solvable in $O(n^2 \log n)$</td>
</tr>
<tr>
<td></td>
<td>Table-based optimal scheduler</td>
<td>Table-based optimal scheduler</td>
<td>EDF optimal</td>
</tr>
<tr>
<td>Sporadic tasks</td>
<td>Simple utilization test</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Optimal fluid scheduler</td>
<td></td>
<td>Pseudo-poly. time solvable with bounded utilization</td>
</tr>
<tr>
<td></td>
<td>(Only implicit deadlines)</td>
<td></td>
<td>EDF optimal (Even arbitrary deadlines)</td>
</tr>
</tbody>
</table>
### Which correctness criteria is the correct one?

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Jobs</th>
<th>Sporadic tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC-1</td>
<td>Polynomial-time solvable (LP formulation)</td>
<td>Simple utilization test</td>
</tr>
<tr>
<td></td>
<td>Table-based optimal scheduler</td>
<td>Optimal fluid scheduler (Only implicit deadlines)</td>
</tr>
<tr>
<td>CC-2</td>
<td>Strongly NP-complete (MILP formulation)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Table-based optimal scheduler</td>
<td></td>
</tr>
<tr>
<td>CC-3</td>
<td>Solvable in $O(n^2 \log n)$</td>
<td>Pseudo-poly. time solvable with bounded utilization</td>
</tr>
<tr>
<td></td>
<td>EDF optimal</td>
<td>EDF optimal</td>
</tr>
<tr>
<td></td>
<td>(Even arbitrary deadlines)</td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion:**

CC-3 $\Rightarrow$ CC-2 $\Rightarrow$ CC-1
What is the modeling overhead?

The modeling overhead of CC-3

Over-approximating either CC-1 or CC-2 by CC-3 has a worst-case speedup cost of 2 (which is tight).
What is the modeling overhead?

The modeling overhead of CC-3

Over-approximating either CC-1 or CC-2 by CC-3 has a worst-case speedup cost of 2 (which is tight).

The modeling overhead of CC-2

Over-approximating CC-1 by CC-2 has a worst-case speedup cost in $[\varphi, 2]$. ($\varphi \approx 1.618$ is the golden ratio.)
∀Thank you!

∃Questions?