Accurate Strategy for Mixed Criticality Scheduling

Yasmina Abdeddaïm

LIGM, Univ Gustave Eiffel, ESIEE Paris

Laboratoire d'informatique Gaspard-Monge (LIGM)

Plan

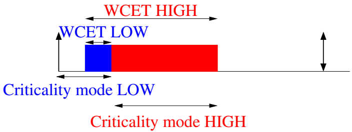
1. Mixed Criticality
2. A Timed Game Model
3. Conclusion

Mixed Criticality Systems

- Applications of different criticality run on the same hardware platform
- Ensure that low criticality applications cannot disturb those of highest criticality
- Use the platform efficiently

Real-time scheduling point of view - Vestal model (2007)

- Level of criticality of tasks: low criticality or high criticality (or more levels)
- A task has different estimates for its worst case execution time (WCET), one per level of criticality
- At the beginning of the execution, the system is in a low criticality mode
- If a high criticality task does not notify its completion after the execution of its low WCET, the system enters the high criticality mode



The Task Set Model

1. Uniprocessor scheduling
2. Two levels of criticality
3. Preemptive job (an instance of a task) level fixed priority scheduling
4. A set of sporadic tasks $\tau_i = (L_i, pr_i, T_i, D_i, C_i)$ with
    - $L_i \in \{LO, HI\}$ the criticality of the task,
    - $pr_i \in \{1 \dots n\}$ the priority of the task,
    - T_i its minimal inter-arrival time,
    - D_i its constrained deadline,
    - For high criticality tasks: $C_i = (C_i(LO), C_i(HI))$ with $C_i(LO) \leq C_i(HI)$,
    - For low criticality tasks: $C_i = C_i(LO)$

Mixed Criticality Scheduling Problem

- In low criticality mode: all the tasks meet their deadlines
- In high criticality mode: all the high criticality tasks meet their deadlines

Standard Mixed Criticality Scheduling Approach

In the high criticality mode no low criticality task is released

Criticism of the Standard Approach

1. If the criticality mode of the system is high it never comes back to low criticality
2. When the system is in high criticality, the execution time of all the high tasks is supposed to be equal to the WCET of the high criticality mode
3. When the criticality mode of the system increases, less critical tasks are definitely no more activated

In This Paper

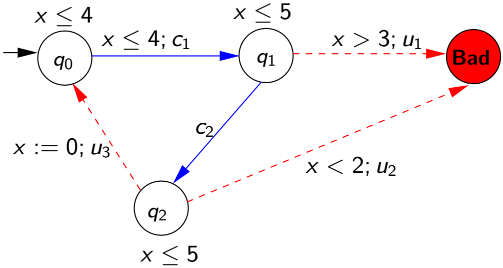
1. When no job exceeds its low criticality WCET the criticality of the system decreases
2. We introduce the notion of criticality configuration of the system that gives the set of jobs exceeding their low WCET **(critical jobs)**
3. The designer can specify the subset of low criticality tasks to stop when the system is in high criticality mode depending on the criticality configuration **(the fault mode policy)**

Adaptive Fault Mode Strategy (AFM)

- A fault mode policy, $policy_i$, is defined by the designer
- A task set is AFM schedulable according to the scheduling algorithm *Sched* and a fault mode policy $policy_i$ if and only if
  1. all the tasks respect their deadlines when the criticality mode of the system is LO
  2. all the active jobs respect their deadlines when the criticality mode of the system is HI and the fault mode policy $policy_i$ is applied

Plan

Timed Game Automata



The set of transitions Δ is composed of:

- Δ_c for controllable transitions: Controller
- Δ_u for uncontrollable transitions: Environment

A Controller plays against the Environment

The goal is to avoid state Bad whatever are the decisions of the environment

Timed Game $G(\mathcal{A}, \mathit{Init}, \phi)$

Timed Game

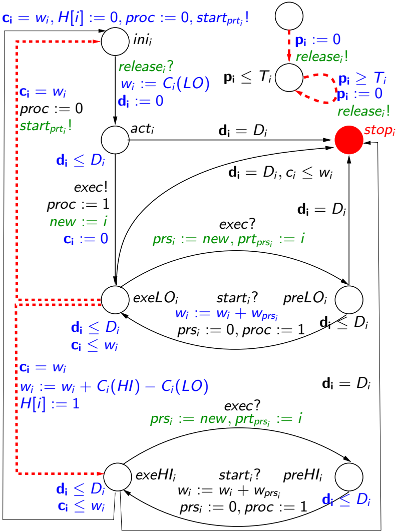
A Timed Game $G(\mathcal{A}, \mathit{Init}, \phi)$ is defined by:

- $\mathcal{A}$ a timed game automaton,
- *Init* initial configurations,
- φ a logic formula.

The timed game $G(\mathcal{A}, \mathit{Init}, \varphi)$ is a Wining Game if there exists a strategy f s.t. the execution of $\mathcal{A}$ from *Init* and supervised by f always satisfies formula φ.

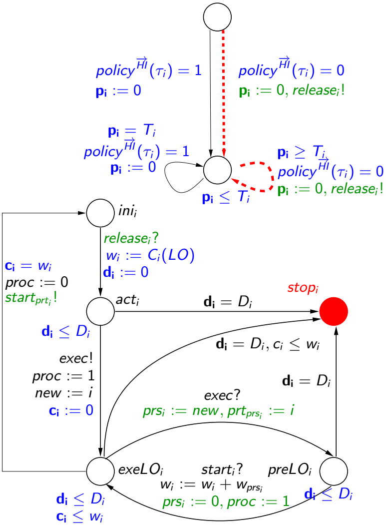
High Criticality Real-Time Task

- Clocks c_i, d_i, p_i
- w_i is the response time of a job of τ_i
- $H[i] = 1$ if a job of task τ_i is critical
- prs_i is the task preempting τ_i
- prt_i is the task preempted by τ_i
- $release_i$ is a signal laughed when a job of τ_i is released

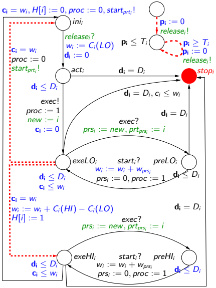


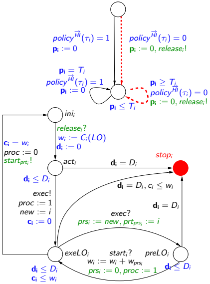
Low Criticality Real-Time Task

- Clocks c_i, d_i, p_i
- w_i is the response time of a job of τ_i
- $policy^{\overrightarrow{HI}}(\tau_i) = 1$ if jobs of τ_i are not activated when the criticality configuration is $\overrightarrow{HI}$
- prs_i is the task preempting τ_i
- prt_i is the task preempted by τ_i
- $release_i$ is a signal laughed when a job of τ_i is released



Exact AFM Feasibility (Job Level Fixed Priority Scheduling)





- Feasible Run: infinite run where no configuration contains a state $stop_i$
- The scheduling problem is feasible if a feasible run exits
- Property to check in the timed game:

$$AG\neg\left(\bigvee_{\tau_i\in\Gamma} stop_i\right) \quad (1)$$

Exact AFM Fixed Priority Schedulability Test

The task set is Fixed Priority schedulable if and only if there exists a run where

- no job misses its deadline (state $stop_i$ not reached)
- a job of a task τ_i cannot be active or preempted if a job of lower priority is executed

$$AG\neg\Big(\bigvee_{\tau_i\in\Gamma} stop_i\Big) \bigwedge \neg\Big(\bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in lp(i)} (act_i \wedge exeLO_j) \bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in lp(i)} (act_i \wedge exeHI_j)$$
$$\bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in lp(i)} (preLO_i \wedge exeLO_j) \bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in lp(i)} (preLO_i \wedge exeHI_j) \qquad (2)$$
$$\bigvee_{\tau_i\in\Gamma^{HI}}\bigvee_{\tau_j\in lp(i)} (preHI_i \wedge exeLO_j) \bigvee_{\tau_i\in\Gamma^{HI}}\bigvee_{\tau_j\in lp(i)} (preHI_i \wedge exeHI_j)\Big)$$

Exact AFM Earliest Deadline First Schedulability Test

The task set EDF schedulable if and only if there exists run where

- no job misses its deadline (state $stop_i$ not reached)
- a job of a task τ_i cannot be active or preempted if a job with an absolute deadline less close to its deadline is executed

$$AG\neg\Big(\bigvee_{\tau_i\in\Gamma} stop_i\Big)\bigwedge\neg\Big(\bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in\Gamma}(act_i\wedge exeLO_j\wedge p_{ij})\bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in\Gamma^{HI}}(act_i\wedge exeHI_j \wedge p_{ij})\bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in\Gamma}(preLO_i\wedge exeLO_j\wedge p_{ij})\bigvee_{\tau_i\in\Gamma}\bigvee_{\tau_j\in\Gamma^{HI}}(preLO_i\wedge exeHI_j\wedge p_{ij}) \bigvee_{\tau_i\in\Gamma^{HI}}\bigvee_{\tau_j\in\Gamma}(preHI_i\wedge exeLO_j\wedge p_{ij})\bigvee_{\tau_i\in\Gamma^{HI}}\bigvee_{\tau_j\in\Gamma^{HI}}(preHI_i\wedge exeHI_j\wedge p_{ij})\Big) \tag{3}$$

p_{ij} is a state of an observer automaton reachable when $d_i - d_j > D_i - D_j$ with d_i and d_j the deadline clocks of tasks τ_i and τ_j respectively

Task set Γ_1

τ_i	L_i	pr_i	T_i	D_i	C_i
τ_1	HI	3	10	10	(1,2)
τ_2	HI	2	8	8	(2,4)
τ_3	LO	1	4	4	(2)

Adaptive Mixed Criticality Strategy (AMC)

- when the criticality of the system is HI:
  - LO criticality tasks are no more activated
  - all jobs of HI criticality tasks are supposed to have an execution time equal to their HI WCET
- we consider that the system returns to LO mode at the first instant where no active job is critical

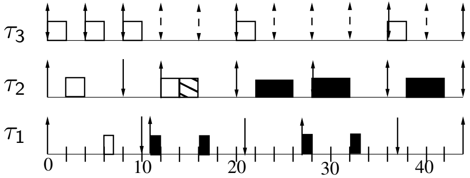


Figure: FP Scheduling of Γ_1 using AMC

Using AMC 6 jobs of τ_3 are not executed

Fixed Priority AFM and $policy_1$

- $policy_1^{\overrightarrow{HI}}(\tau_3) = 1$ for all the criticality configurations where τ_2 has a critical job
- $policy_1^{\overrightarrow{HI}}(\tau_3) = 0$ otherwise

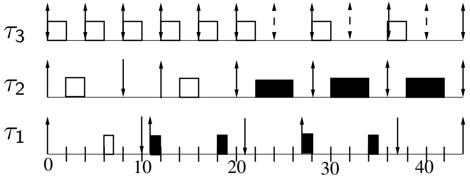


Figure: FP Scheduling of Γ_1 using AFM and $policy_1$

Using Fixed Priority AFM and $policy_1$ 3 jobs of τ_3 are not executed

Fixed Priority AFM and *policy*$_2$

- $policy_2^{\overrightarrow{HI}}(\tau_3) = 1$ for all the criticality configurations where "only" τ_2 has a critical job
- $policy_2^{\overrightarrow{HI}}(\tau_3) = 0$ otherwise

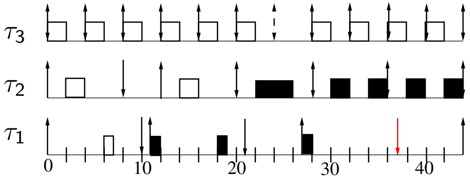


Figure: FP Scheduling of Γ_1 using AFM and *policy*$_2$

Not Fixed Priority AFM schedulable using *policy*$_2$

EDF AFM and *policy*$_2$

- $policy_2^{\overrightarrow{HI}}(\tau_3) = 1$ for all the criticality configurations where "only" τ_2 has a critical job
- $policy_2^{\overrightarrow{HI}}(\tau_3) = 0$ otherwise

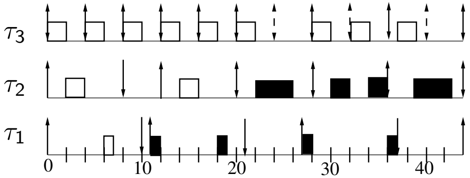


Figure: EDF Scheduling of Γ_1 using AFM and *policy*$_2$

LO criticality jobs are necessary in some HI criticality configurations

Plan

In this work

- A scheduling strategy for the mixed criticality real-time scheduling problem where **the designer can define his own policy to deal with low criticality tasks**
- **Exact** feasibility and schedulability tests for job level fixed priority algorithms based on CTL model checking for timed game automata
- **Future work**
  - propose a more specific game model taking into account the characteristics of our real-time scheduling problem to reduce the state space explosion problem
  - generate the fault mode policy