ETR, September 2026

Simulation-Based Scheduling Analysis: Theoretical Problems & Practical Challenges

Hai Nam Tran

From collaborations with Frank Singhoff, Stéphane Rubini, Jalil Boukhobza, Blandine Djika, Alan Le Boudec, Sebastien Levieux, Damien Guidolin-Pina, Emmanuel Grolleau

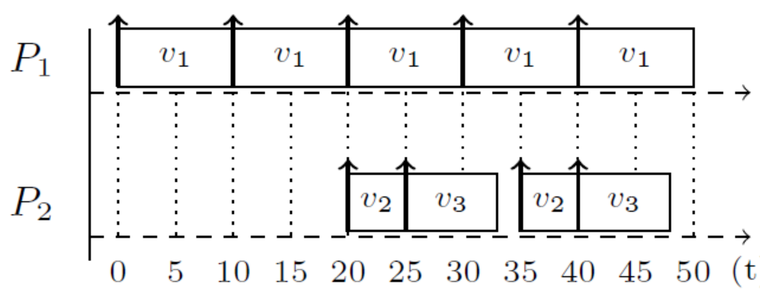
Real-time Scheduling Analysis

1. Define, for a given system

- **Architecture model**
  - Software: tasks, shared resources,…
  - Hardware: processing units (processor, core), memory hierarchy,…
  - Deployment: which tasks on which processing units/resources
- **Scheduler**
  - Scheduling algorithms/policies

2. Schedulability Tests

Focus of the lecture

WCRT Analysis	Scheduling Simulation
$R_i = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \cdot C_j$	

Scheduling Simulation

Definition

Construction of an execution trace of a task set T on an architecture M under a scheduler S over an interval of time F

- **Example**

Abstract timing model of a task

  - $\boldsymbol{T}$: **a set of periodic tasks** $\tau_i(C_i, T_i, D_i, \Pi_i, O_i)$ **- capacity, period, deadline, priority and offset**
  - $\boldsymbol{M}$: **uniprocessor**
  - $\boldsymbol{S}$: **fixed priority non-preemptive scheduling**
  - $\boldsymbol{F} = $ **[0, Hyperperiod)**

	O_i	C_i	D_i	T_i	Π_i
τ_1	0	1	3	3	3
τ_2	0	2	6	6	2
τ_3	0	4	12	12	1

Task parameters from (Richard, 2005)

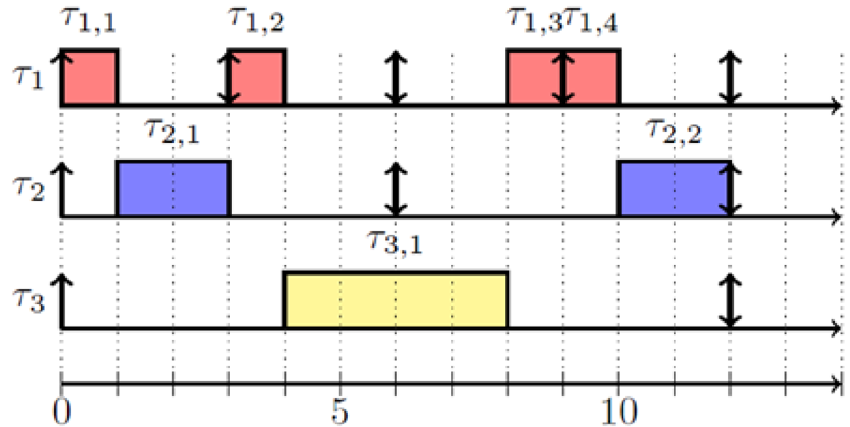
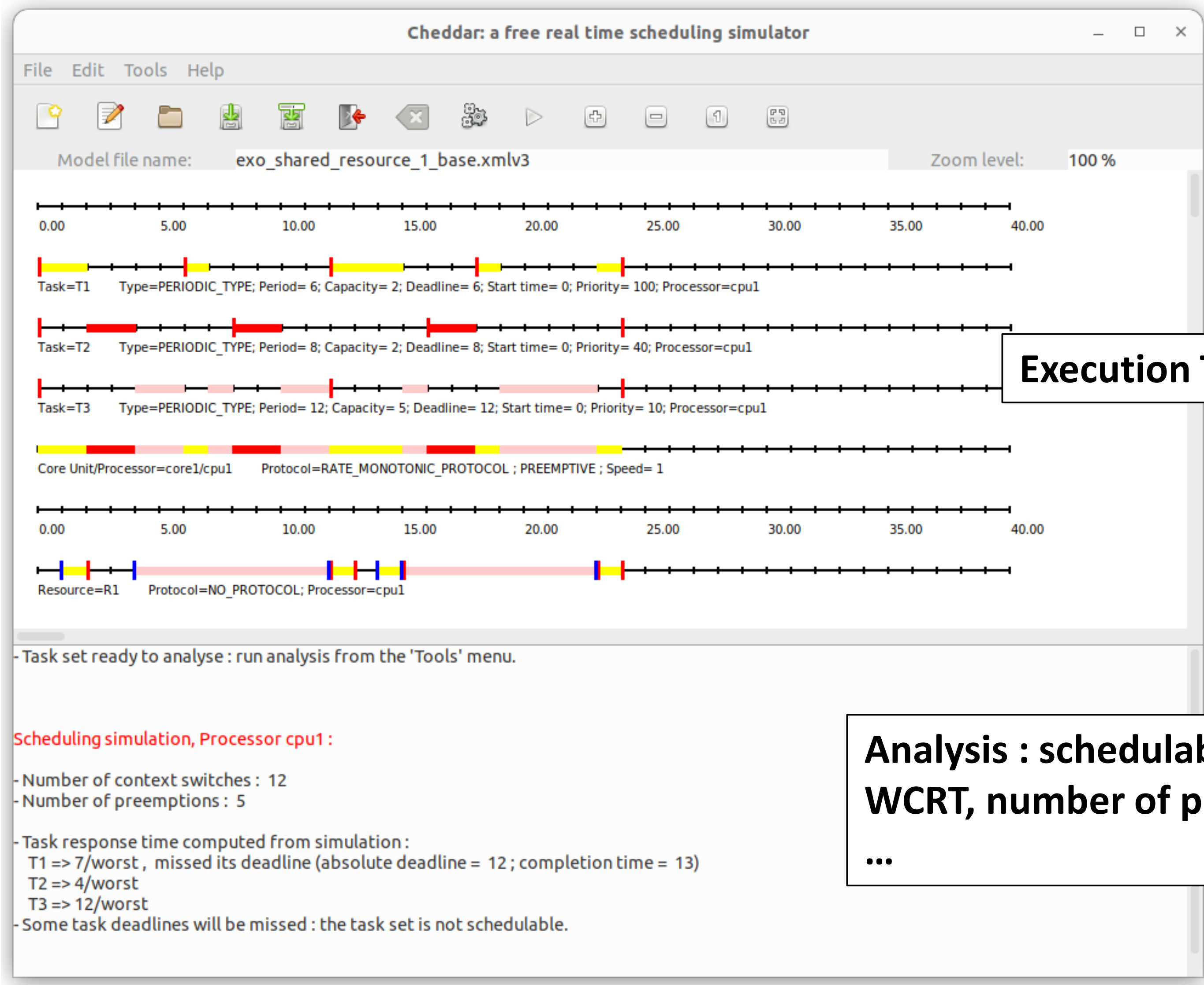


Figure taken from B. Djika's PhD Thesis

Scheduling Simulation Example



Execution Trace

Analysis : schedulability, WCRT, number of preemption …

Scheduling Simulation

- **Capacity parameter - C_i**
  - **An input to the model**
  - **Obtained by WCET analysis**
  - **WCET & Interference**

WCET OTAWA
Thursday 10 - 15:30 - 17:30

In this lecture, **interference** is defined as delays caused by contention for shared hardware resources: cache, memory bus, DRAM, interconnect, …

Isolation WCET + Explicit Interference	Interference-aware WCET
• The simulator performs scheduling simulation and compute interference • Require an **interference computation model C**	• The simulator does not account for interference in the simulation

There is no ideal approach, always specify explicitly your definition of WCET

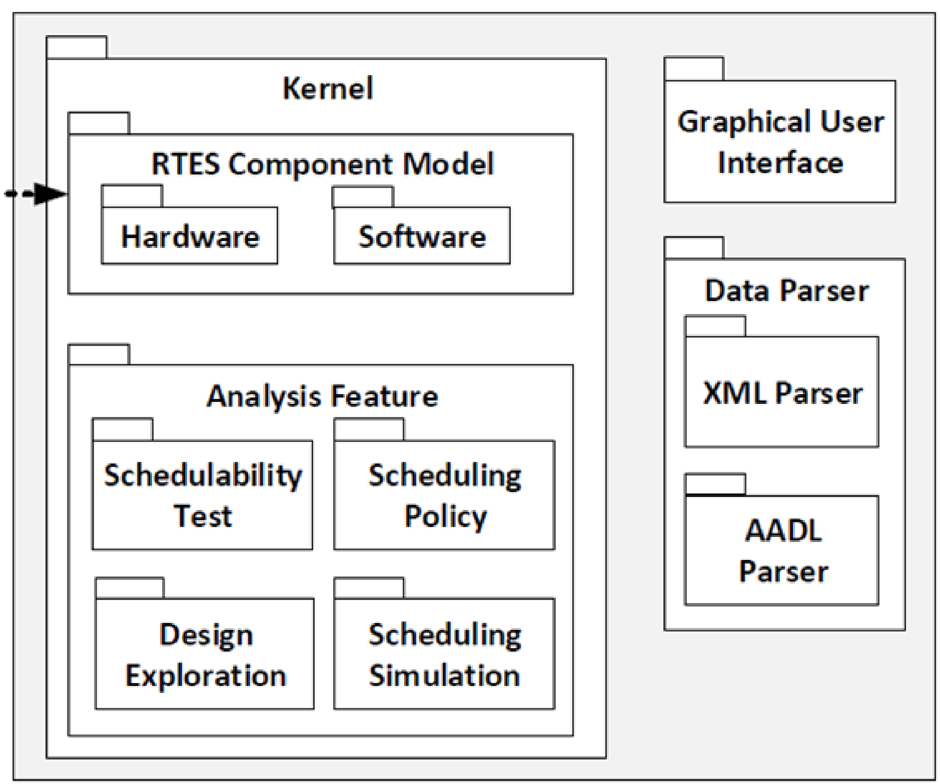
Advantages & Limitations

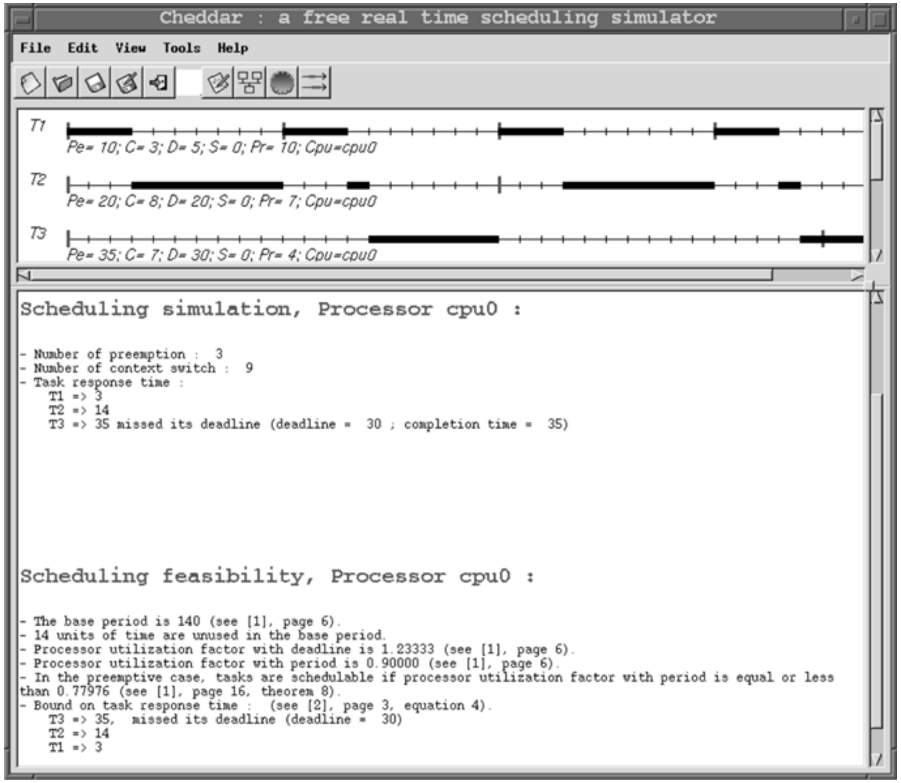
- **A common practice of actors in the real-time community**

Advantages	Limitations
• **Observed reduced pessimism compared to WCRT analysis** - Used to as a baseline to compare analyses • **Adaptability/Flexibility** - Integration of additional scheduling parameters • **Observability** - Record and analyze properties such as numbers of preemptions, total preemption costs, and various scheduling events that are not observable by static analysis • **Analysis** - Understand why a system is not schedulable	• **Scalability** - Especially when mixing timing specifications of different orders of magnitude; e.g: WCET and cache block reload time. • **Analysis** - Tons of trace • **Engineering challenge** - Need to implement the simulator • **Theoretical problems** - Sustainability / Scheduling Anomaly - Simulation Interval

Cheddar Scheduling Analyzer

- **Developed and maintained by Lab-STICC, Univ. Brest**
  - **Started in 2002 by Frank Singhoff**
  - **Since 2008, integration in AADLInspector, Ellidiss Technologies**
  - **Since 2011, integration in TASTE toolchain of ESA**





An Example of Use: PLATO/ESA (2021-2022)

- **PLATO: PLAnetary Transits and Oscillation of Stars**
  - **Verify the data processing unit of a satellite**
  - **Used during the CDR (Critical Design Review)**
- **About 1.1k lines of AADL model**
  - **Multi-core partitioned scheduling**
  - **12x LEON cores (GR712RC)**
  - **24 cameras**
  - **34 threads**
  - **107 AADL components**
  - **560 AADL property associations**
  - **Required Cheddar kernel update and a new AADL Inspector version (AMP multicore scheduling analysis)**
- **Launch is planned for fall 2026**

Sebastien Levieux's PhD Thesis

A Safe Schedulability Test ?

- Several questions to answer

Does the simulated scenario repres... e worst case ?

Sustainability & Scheduling Anomaly

If the system is schedulable in the worst case, does it ... ins schedulable in better cases ?

How long should we simulate ?

Simulation Interval

Does the system model capture enough the complexity of actual hardware ?

Sustainability & Scheduling Anomaly

Sustainability

Sustainability (Goossens et al., 1997)
A given scheduling policy and/or a schedulability test is sustainable if any system that is schedulable under its worst-case specification remains so when its behavior is better than the worst-case

- **The term "better" means that the parameters of one or more individual task(s) are changed in any, some, or all of the following ways**
  - **(1) reduced capacity**
  - **(2) larger period**
  - **(3) larger relative deadline**

Scheduling Anomaly

Timing anomaly (Lundqvist and Stenström, 1999)

A timing anomaly is a situation where the local worst-case does not entail the global worst-case

Scheduling anomaly (Richards, 1960 and Graham, 1966)

An anomaly is a situation in which the total run-time can be increased by

1. **Increasing the number of processors**
2. **Shortening the run-time of some tasks**
3. **Changing the order of the task list**
4. **Removal of some precedence relations**

Scheduling anomaly (Almeida et al., 2006)

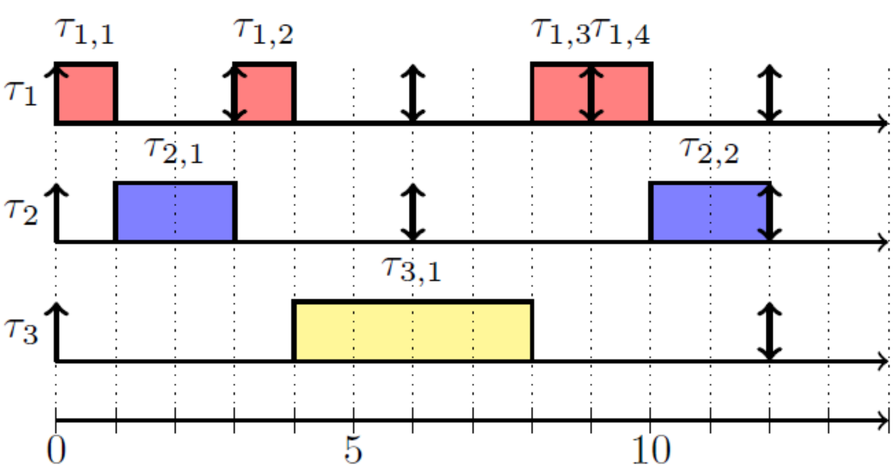
The expression *scheduling anomaly* in real-time scheduling is used to refer to a counter intuitive phenomenon in which increasing the systems resources or relaxing the application constraints can lead to increased schedule length

Reduced Capacity

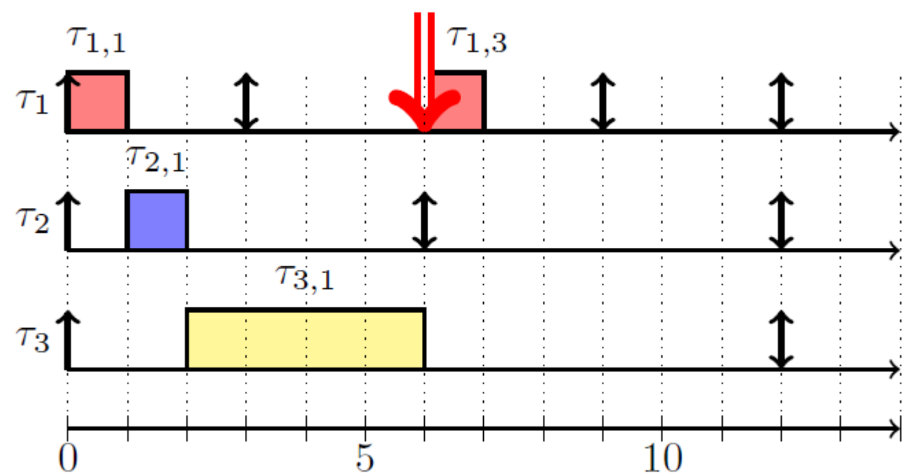
Task parameters from (Richard, 2005)

	O_i	C_i	D_i	T_i	Π_i
τ_1	0	1	3	3	3
τ_2	0	2	6	6	2
τ_3	0	4	12	12	1

Mono-processor, independent, synchronous tasks
Fixed-priority non-preemptive scheduler



$C_2 = 2$, **schedulable** system



$C_2 = 1$, **not schedulable**

Figures taken from B. Djika's PhD Thesis

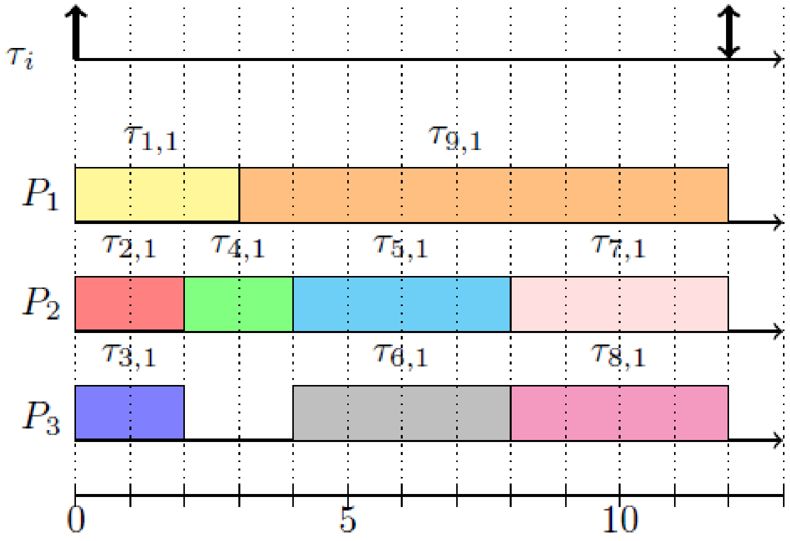
Reduced Capacity

Task parameters from (Graham, 1966)

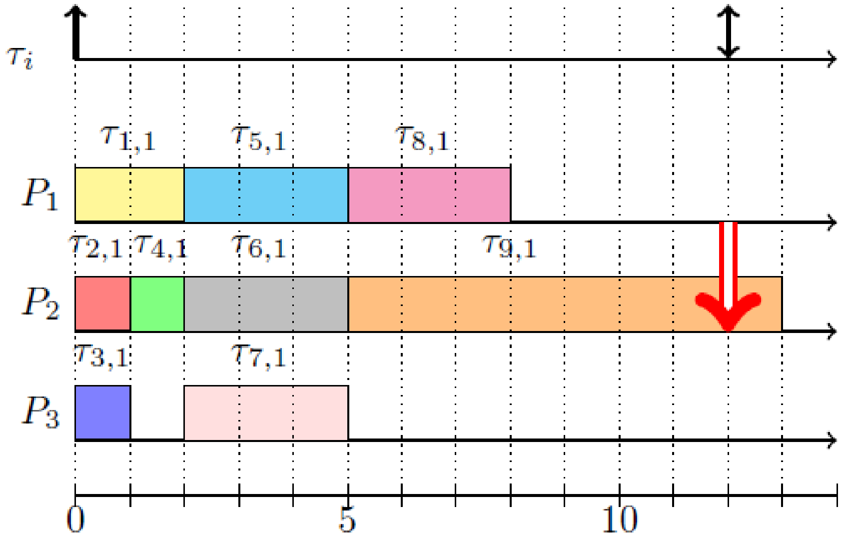
	O_i	C_i	D_i	T_i	Π_i
τ_1	0	3	12	12	9
τ_2	0	2	12	12	8
τ_3	0	2	12	12	7
τ_4	0	2	12	12	6
τ_5	0	4	12	12	5
τ_6	0	4	12	12	4
τ_7	0	4	12	12	3
τ_8	0	4	12	12	2
τ_9	0	9	12	12	1

Multi-processor (3), synchronous tasks
Fixed-priority non-preemptive scheduler
Task migration
Dependencies: $\tau_1 \prec \tau_9, \tau_4 \prec \tau_5, \tau_4 \prec \tau_6, \tau_4 \prec \tau_7, \tau_4 \prec \tau_8$

Reduced Capacity



Schedulable system



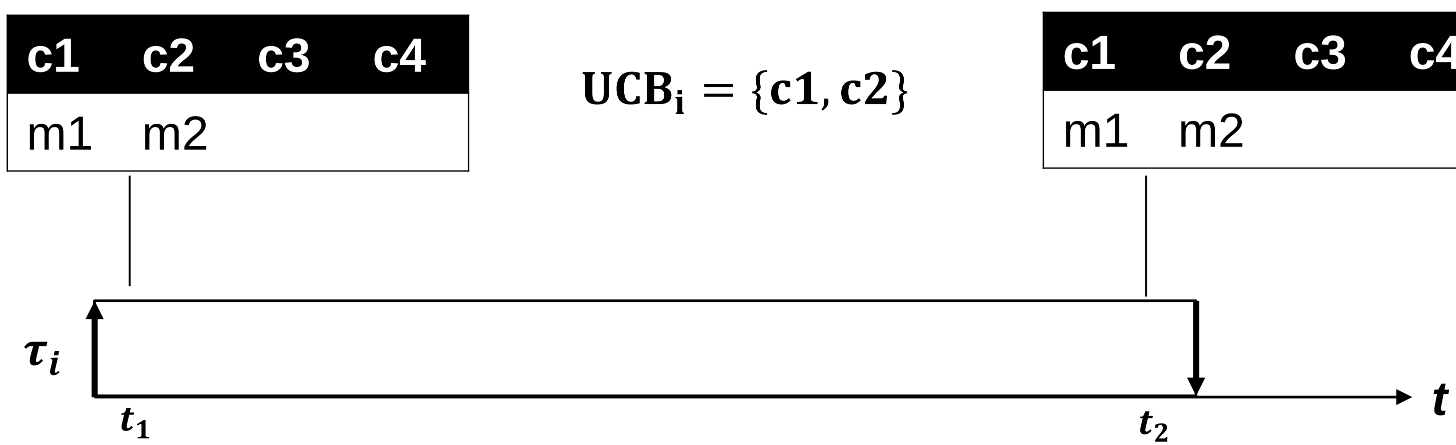
All capacities reduced by 1,
System not schedulable

Figures taken from B. Djika's PhD Thesis

Interference Problem: Example with CRPD

Cache related preemption delay (CRPD)
The additional time to refill the cache with memory blocks evicted by preemption

Useful Cache Blocks - UCB (Lee et al., 1998)
Cache blocks that hold memory blocks, which are: (1) cached and (2) reused without eviction



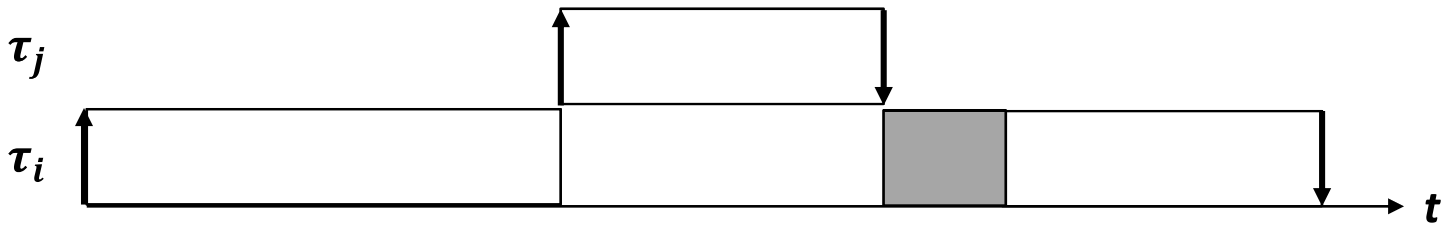
Interference Problem: Example with CRPD

Evicting Cache Blocks - ECB (Busquets et al., 1996)

Cache blocks that are accessed/placed in the execution of the preempting task.

$$\mathbf{ECB_j = \{c1, c2, c3\}}$$

c1	c2	c3	c4
m5	m6	m7	



$$\mathbf{CRPD = BRT\,.\,|UCB_i - UCB_i'|}$$

Remaining UCB in the cache

Block Reload Time

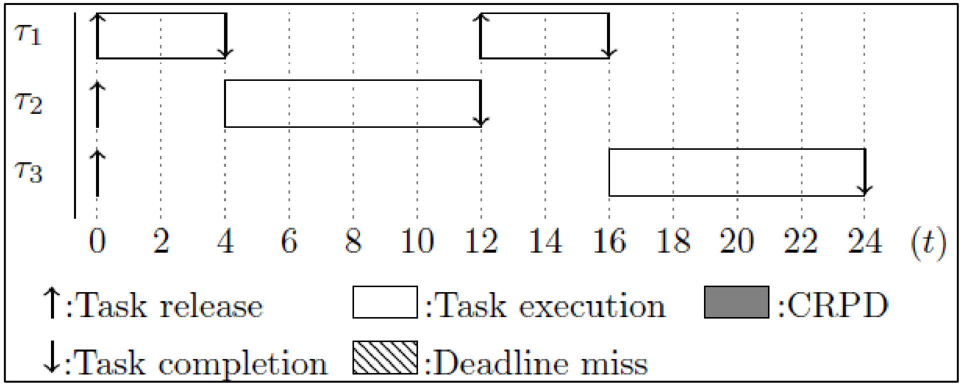
Set of UCB of τ_i

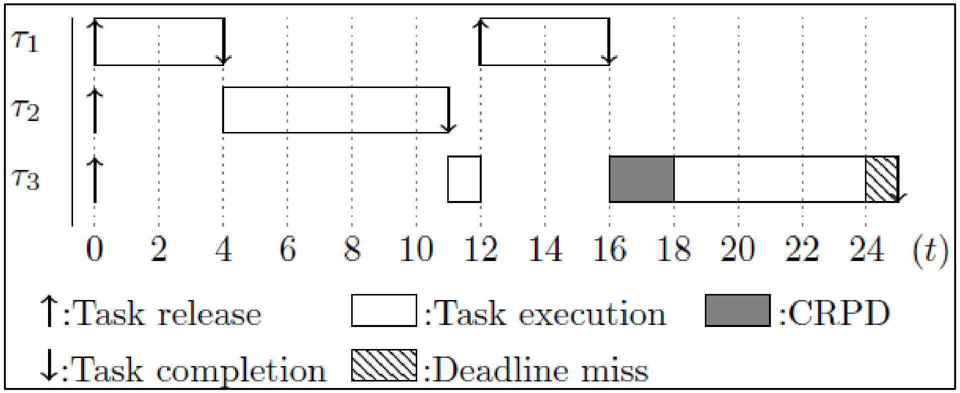
Interference Problem: Example with CRPD

- **Sustainability Analysis**
  - **Example 1**
    - Task set $T = \{\tau_1, \tau_2, \tau_3\}$
    - $BRT = 1$
  - **Case 1: original task set**
    - Schedulable, all deadlines are met
  - **Case 2: reduced capacity of τ_2**
    - $C'_2 = 7\ (< C_2 = 8)$
    - $\gamma_3^{16} = \left|UCB_3 - UCB_3^{16}\right| \cdot BRT$
      $= |\{1,2\} - \emptyset| = 2$
    - τ_3 missed its deadline
    - **Non sustainable scheduling with regard to the capacity parameter**

Phavorin et al., 2015

Task	C_i	T_i	D_i	O_i	Π_i	$\mathrm{UCB_i}$	$\mathrm{ECB_i}$
τ_1	4	12	12	0	3	$\emptyset$	{1,2}
τ_2	8	24	24	0	2	{3}	{3,4}
τ_3	8	24	24	0	1	{1,2}	{1,2}



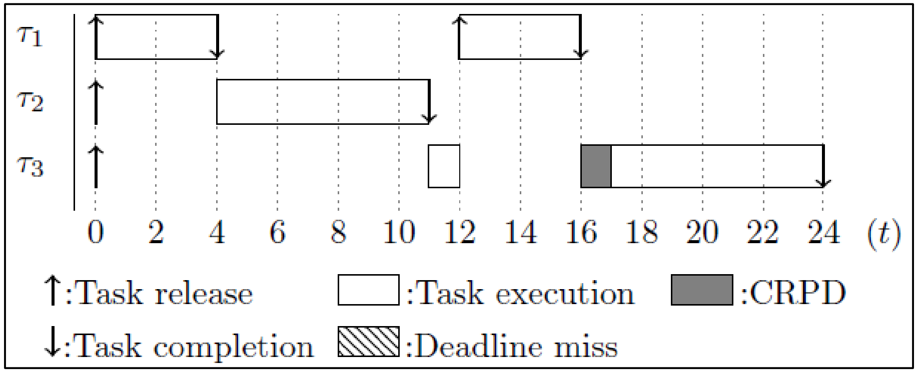


Interference Problem: Example with CRPD

- **Sustainability Analysis**
  - **Example 1**
    - Task set $T = \{\tau_1, \tau_2, \tau_3\}$
    - $BRT = 1$

Task	C_i	T_i	D_i	O_i	Π_i	$\mathrm{UCB_i}$	$\mathrm{ECB_i}$
τ_1	4	12	12	0	3	$\emptyset$	{1,2}
τ_2	8	24	24	0	2	{3}	{3,4}
τ_3	8	24	24	0	1	{1,2}	{1,2}

  - **Case 3: Reduced capacity of τ_2**
    - $C'_2 = 7\ (< C_2 = 8)$
    - CRPD limited by executed capacity
    - Schedulable task set



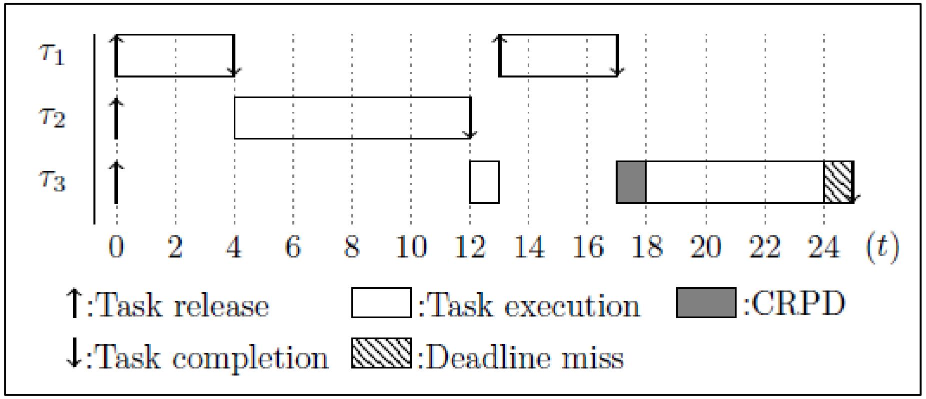
Tran et al., 2021

Interference Problem: Example with CRPD

- **Sustainability Analysis**
  - **Example 1**
    - Task set $T = \{\tau_1, \tau_2, \tau_3\}$
    - $BRT = 1$

Task	C_i	T_i	D_i	O_i	Π_i	$\mathrm{UCB_i}$	$\mathrm{ECB_i}$
τ_1	4	12	12	0	3	$\emptyset$	{1,2}
τ_2	8	24	24	0	2	{3}	{3,4}
τ_3	8	24	24	0	1	{1,2}	{1,2}

  - **Case 4: Larger period**
    - $T_1' = 13 > T_1 = 12$
    - CRPD limited by executed capacity
    - **τ_3 missed its deadline**



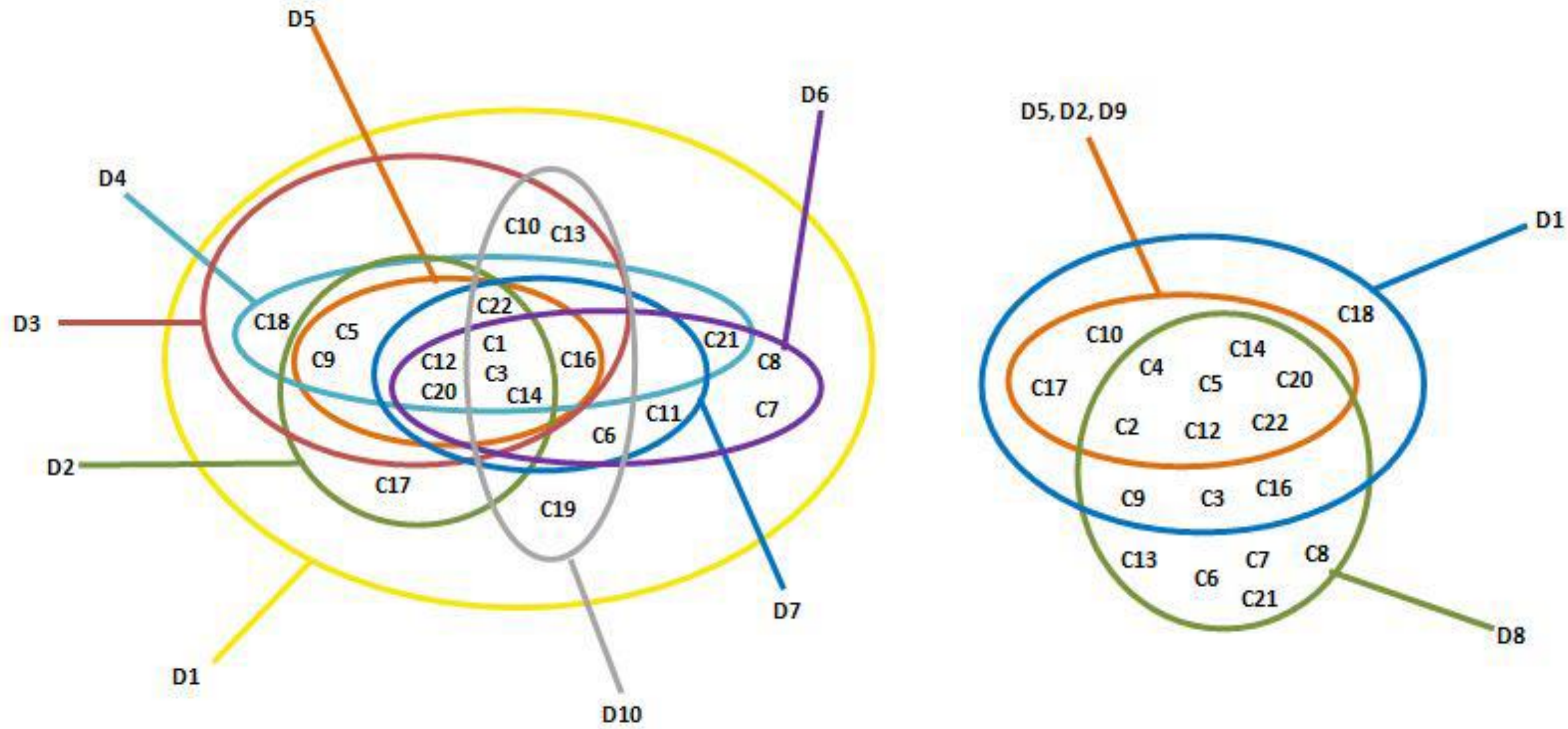
Tran et al., 2021

Scheduling Anomalies Detection

- **Contribution: Model to investigate conditions leading to scheduling anomalies**
  - **Static constraints (C1-C23): processing unit, scheduling type, scheduling policy, priority assignment, preemptivity, release time, task type, dependency, self-suspending, deadline constraints**
  - **Dynamic constraints - Scheduling anomalies (D1-D10)**
    - Task execution is shorter than expected
    - Dependency constraints are not met
    - Processor speeds have been increased
    - …

Can we design a model to identify scheduling anomalies? Blandine Djika et al. 14th International Real-Time Scheduling Open Problems Seminar (RTSOPS 2025), Jul 2025, Bruxelle, Belgium.

Scheduling Anomalies Detection



Set oriented representation for all considered scheduling anomalies. The **left** graph is representing anomalies for **uniprocessor platforms** while the **right** one is for **multiprocessor**.

Can we design a model to identify scheduling anomalies? Blandine Djika et al. 14th International Real-Time Scheduling Open Problems Seminar (RTSOPS 2025), Jul 2025, Bruxelle, Belgium.

ETR, September 2026

Simulation Interval

Simulation Interval

- **How long should we run the simulation ?**

Feasibility Interval (Goossens et al., 1997)
A finite interval such that if all the deadlines of jobs released in the interval are met, then the system is schedulable

  - **The feasibility interval includes the worst case schedule states regarding schedulability**

Simulation Interval
A finite and safe time interval such that the schedule start to repeat in a cycle

  - **The simulation interval includes all possible (reachable) schedule states**

Simulation Interval

Reference	Assumptions	Simulation Interval UB
Leung et al., 1982	$D_i \leq T_i$, asynchronous, independent, mono-processor, EDF or FTP scheduler	$\boldsymbol{O_{max} + 2H}$
Goossens et al., 1997	$D_i \leq T_i$, asynchronous, independent, mono-processor, FTP scheduler	$\boldsymbol{S_n + H}$
Goossens et al., 1999	$D_i \neq T_i$, asynchronous, independent, mono-processor, FTP scheduler	$\boldsymbol{O_{max} + 2H}$

S_n : Stabilization time

Assume that tasks are ordered by priority level

$$\boldsymbol{S_1 = O_1}$$

$$\boldsymbol{S_i = max(O_i, O_i + \left\lceil \frac{S_{i-1} - O_i}{T_i} \right\rceil)}$$

Simulation Interval

Choquet-Geniet et al., 2004	$D_i \leq T_i$, asynchronous, resource-sharing tasks with precedence constraints, mono-processor, work conserving scheduler
Simulation interval UB : $\boldsymbol{\theta} + \boldsymbol{H} \leq O_{max} + 2H$ θ: latest acyclic idle time $\theta \leq O_{max} + H$	
Note: Finding θ requires storing every idle time position until the first window of size H containing exactly $H(1-U)$ idle time units is found	

Simulation Interval

Goossens et al., 2022	$D_i \neq T_i$, asynchronous, work-conserving deterministic and memory-less scheduler, with preemption delay

Simulation interval for general cases

$$\boldsymbol{H \cdot (n+1) \cdot (\alpha_{max}+1) \times \prod_{i=1}^{n} ((O_i + D_i - T_i)_0 + 1)}$$

α_{max} : the preemption cost

The authors also state that :

- For EDF, the Simulation interval UB = $\mathrm{O_{max}} + 2\mathrm{H}$
- For FTP scheduler, the Simulation interval UB = $S_n + H$

Simulation Interval

Guidolin-Pina et al., 2026	$D_i \leq T_i$, asynchronous, work-conserving deterministic and memory-less scheduler
Simulation interval UB: $\boldsymbol{t_0} + \boldsymbol{H}(\boldsymbol{1} + \boldsymbol{U}) \leq O_{max} + 2H$ t_0: is the instant just before the set of clocks behave cyclically $t_0 = \max(0^-, \max(\mathrm{O_i} - \mathrm{T_i}))$	
Note: only need to simulate up to the first idle point strictly after $t_0 + H$	

Simulation Interval in Scheduling Analysis Tool

- Since the early 2000s, several scheduling analysis tools have been developed
  - SimSo (Chéramy et al., 2014)
  - STORM (Urunuela et al., 2010)
  - Xoncrete (Brocal et al., 2010)
  - MAST (Harbour et al., 2001)
  - TIMES (Amnell et al., 2003)
  - Yartiss (Chandarli et al., 2012)
  - RealTss (Diaz et al., 2007)
  - Cheddar (Singhoff et al., 2002)
- Most of them do not support automated cyclicity analysis or systematic computation of simulation interval bounds

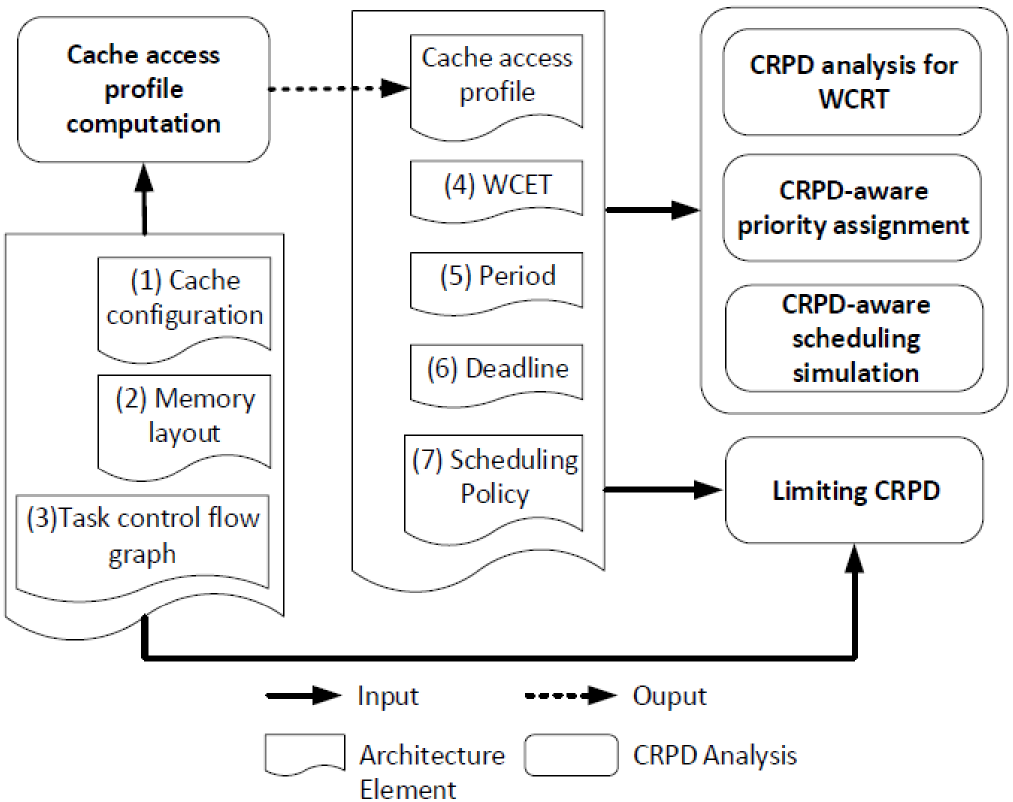
Practical Challenges

Performance

- **Mixing timing specification of different orders of magnitude**
  - **Cache block reload time (μs) is significant smaller than period or capacity of a task (ms).**
  - **With cache, 1 unit of time = 1 μs**
  - **Significantly large scheduling simulation interval needed to verify the schedulability of a task set**
- **Exporting all scheduling events is very I/O intensive**
  - **Running_Task**
  - **Start_Of_Task_Capacity**
  - **End_Of_Task_Capacity**
  - **Allocate_Resource**
  - **Wait_For_Resource**
  - **…**

Tool Interoperability

- **Scheduling simulation tool requires results provided by other analysis tools**
  - **Example : in CRPD analysis, task control flow graph is required to compute cache access profile (UCB_i, ECB_i)**



System Model

	O_i	C_i	D_i	T_i	Π_i
τ_1	0	1	3	3	3
τ_2	0	2	6	6	2
τ_3	0	4	12	12	1

L_i	$C_i(HI)$	$C_i(LO)$
HI		
HI		
LO		

$\prec$
τ_2
τ_3
τ_4

UCB_i	ECB_i

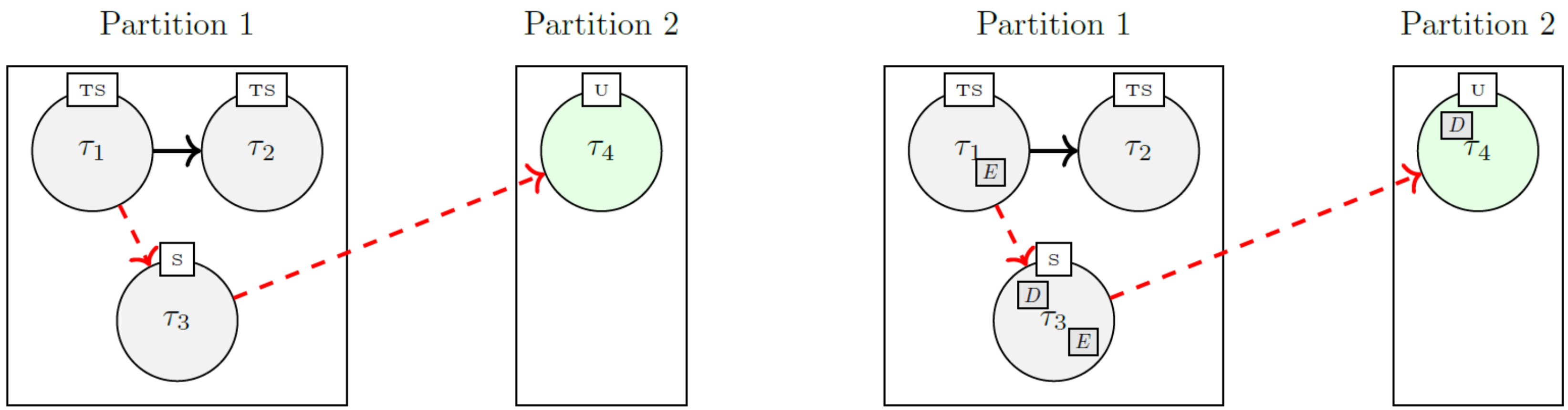
Q_i	….

…	…

More analysis = more complex model

Example: Joined scheduling and security analysis

- **Context: Time and Space Partitioned (TSP) systems - ARINC653**
  - **Trade-offs between schedulability and security**
    - Addition of encryption/hash tasks = **overhead**



Example: Partitioning and communications without/with security functions.
Confidentiality level: TS - Top Secret, S - Secret, U - Unclassified

  - **Scheduling problem: Tasks to partitions assignments**

Atchadam et al. "When security affects schedulability of TSP systems: trade-offs observed by design space exploration." *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).*

Example: Mixed Criticality Scheduling Simulation

- **Context: Scheduling simulation of Mixed Criticality Systems**
  - **Need to handle more parameters than usual scheduling policies**
    - Number of modes
    - Adaptation algorithm
    - Recovery algorithm
  - **Many of those parameters have to be randomly generated for synthetic task sets and also case studies**
    - WCETs/Mode, Period
    - Overrun probability
    - Recovery
  - **Additional evaluation metrics**
    - Time spent in HI or LO mode
    - Number of discarded jobs, mode changes, missed deadlines…

Le Boudec et al. "Poster: Reusable software components to prototype and evaluate mixed-criticality scheduling policies." *IEEE 31st International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2025.

System Model

	O_i	C_i	D_i	T_i	Π_i
τ_1	0	1	3	3	3
τ_2	0	2	6	6	2
τ_3	0	4	12	12	1

L_i	$C_i(HI)$	$C_i(LO)$
HI		
HI		
LO		

$\prec$
τ_2
τ_3
τ_4

UCB_i	ECB_i

Q_i	….

…	…

Need to apply a **model-driven engineering** approach otherwise the model becomes quickly unmanageable.

Usage

Teaching	Research

Desktop Application

Web Application

Command-line Interface

Interactive System Builder

Import/Export Model

Conclusion

Conclusion

- **Questions to answer when scheduling simulation is used in experiments**

Does the simulated scenario represent a worst case ?

If the system is schedulable in the worst case, does it remains schedulable in better cases ?

How long should we simulate ?

Does the system model capture enough the complexity of actual hardware ?

  - Do not always need to prove simulation as a safe schedulability test
    - Can be used as a baseline

    or

    - A proof of unschedulability

Conclusion

- **Cheddar scheduling analyzer**
  - **Open-source, supported by maintained by Lab-STICC, Univ. Brest**
  - **Fall 2026**
    - Release of Cheddar 3.4
    - Pre-alpha releases of Cheddar CLI, Cheddar Web

https://beru.univ-brest.fr/cheddar/

  - **Usage in teaching, research, experiment,… do not hesitate to contact us !**
  - **A Cheddar tutorial (ETR 2021) is available at**

http://beru.univ-brest.fr/cheddar/contribs/educational/ubo/ETR-2015-2017-2021/tp-gb.html