Interference modelling and analysis in hard real-time multiprocessor systems

Patricia Balbastre

Universitat Politècnica de València, Spain

Outline

- Introduction
- Related work
- Task model
- Interference-aware scheduling
- Schedulability analysis
- Task allocation
- Evaluation
- Conclusions

Introduction

- Scheduling of multi core real-time systems introduces more complexity than that of single core systems
- Execution of tasks no longer depends only on their own computation time or that of the highest-priority tasks.
- Shared hardware resources between processors introduce an important pessimism factor →interference
- Three main sources of interferences:
  - Main memory
  - Memory bus
  - Cache

How to measure interference?

Two approaches

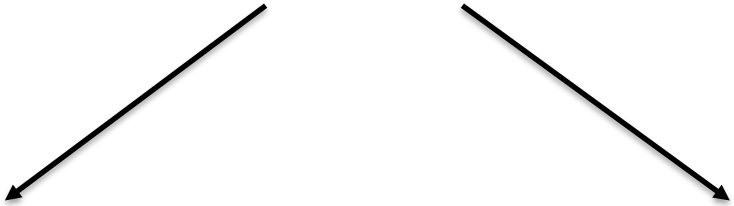
Specific model

- Ad-hoc calculation for a specific resource type
- Only valid for the selected hardware
- Interference value very close to reality

General model

- When hardware vendor does not provide details
- The interference does not depend on the hardware
- But: more pessimistic

How to take into account interference?



Add this value to WCET

➔traditional sched analysis is valid

➔ Very pessimistic

Add this value to the model as a new parameter

➔ Need to define new sched theory

Related work

- Two surveys
  - Until 2018➔ Maiza C. et al (2019) A survey of timing verification techniques for multi-core real-time systems. ACM Comput Surv 52(3)
  - Until 2021➔ Lugo T. et al (2022) A Survey of Techniques for Reducing Interference in Real-Time Applications on Multicore Platforms. IEEE Access 10
- Some examples
  - WCRA parameter (Worst Case number of shared Resource Accesses) (J. Galizzi et al., 2014)
  - isWCET parameter (interference-sensitive Worst Case Execution Time) (Nowotsch et al., 2014)
  - DRAM modeling (Kim et al., 2014)
  - Shared cache modeling (Guo et al., 2020)
  - MRSS model (Davis et al., 2021)

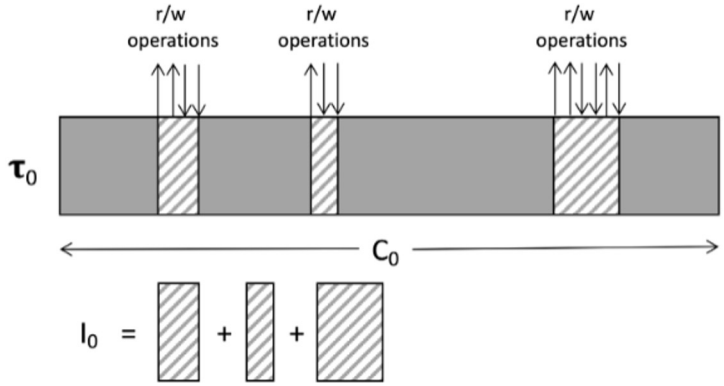
Our proposal

- We propose a general model where interference is a new parameter of the task model
- Highly critical real-time systems
- Dynamic priorities
- Need to:
  - Allocate tasks to cores using interference information
  - Define how this model is scheduled
  - Provide schedulability analysis

Task model

- M cores: M_1, M_2,…, M_m
- N tasks: $\tau = [\tau_1,\tau_2,\ldots,\tau_n]$
- Each task is characterized by $\tau_i = (C_i, D_i, T_i, I_i)$

Interference parameter I_i



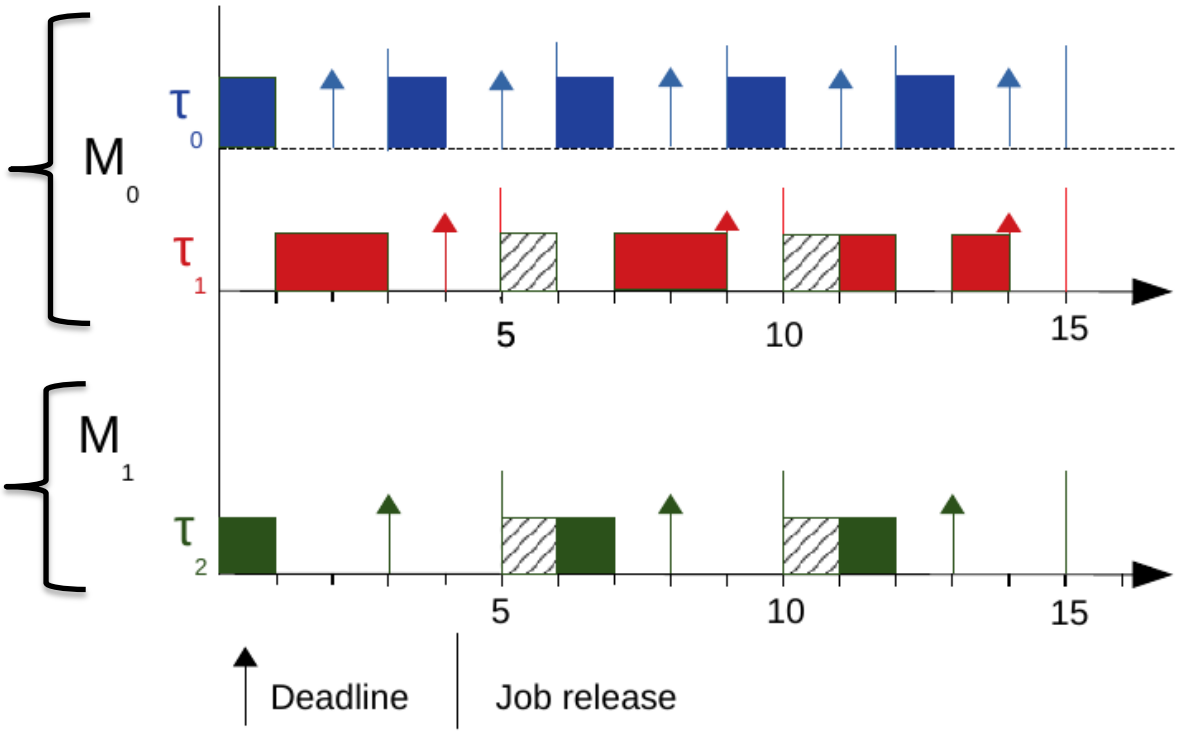
- Broadcasting task τ_i ➔ Provokes interference, $I_i \neq 0$
- Receiving task τ_i ➔ Receives interference so:
  - $I_i \neq 0$
  - there is at least τ_j in another core whose $I_j \neq 0$

Interference-aware scheduling

- Example

$\tau_0 = (1,2,3,0)$
$\tau_1 = (2,4,5,1)$

$\tau_2 = (1,3,5,1)$



Schedulability analysis

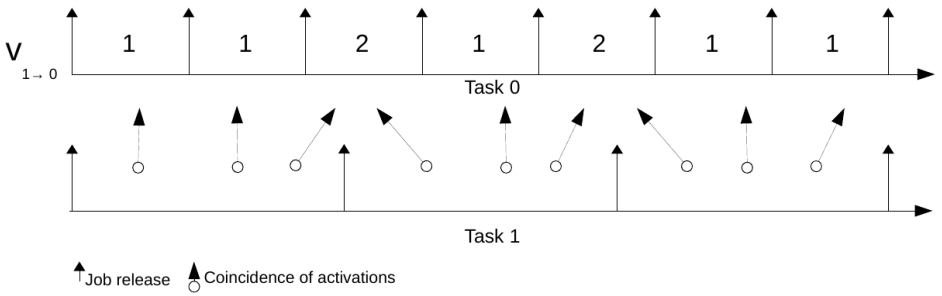
- Dynamic priorities (EDF)
  - Well-known analysis based on demand bound function (dbf)
  - We want to obtain the equivalent dbf for the new model
  - Max. number of activations that τ_j provokes to τ_i

$$\overrightarrow{v_{j\to i}}[a] = 1 + \sum_{t=aT_i+1}^{(a+1)T_i-1} g(t)$$

being

$$g(t) = \begin{cases} 1 & \textit{If } t - T_j \left\lfloor \frac{t}{T_j} \right\rfloor = 0 \\ 0 & \textit{Elsewhere} \end{cases}$$

  - Example:



Schedulability analysis

- Use $v_{j \to i}$ to obtain a modified dbf that incorporates interference
- First approximation: Add $v_{j \to i}$ to WCET → dbf'

$$\max_{0 \le a \le A_i - 1} \overrightarrow{v_{j \to i}}[a] \cdot I_j$$

$$\downarrow$$

$$C'_i = C_i + \sum_{\tau_j \notin M_k} \max_{0 \le a \le A_i - 1} \overrightarrow{v_{j \to i}}[a] \cdot I_j$$

$$\downarrow$$

$$dbf'_{\tau_{M_k}}(t) = \sum_{\forall \tau_i \in M_k} C'_i \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor \longrightarrow dbf'_{\tau_{M_k}}(t) \le t \quad \forall t \le L_b$$

Very pessimistic

Schedulability condition

Schedulability analysis

- Use $v_{j \to i}$ to obtain a modified dbf that incorporates interference
- Second approximation: Add $v_{j \to i}$ to dbf → dbf''

$$dbf''_{\tau_{M_k}}(t) = \sum_{\tau_i \in M_k} \left(C_i \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor + \sum_{\tau_j \notin M_k} \sum_{a=0}^{\left\lfloor \frac{t+T_i-D_i}{T_i} \right\rfloor - 1} \overrightarrow{v_{j \to i}}[a] \cdot I_j \right)$$

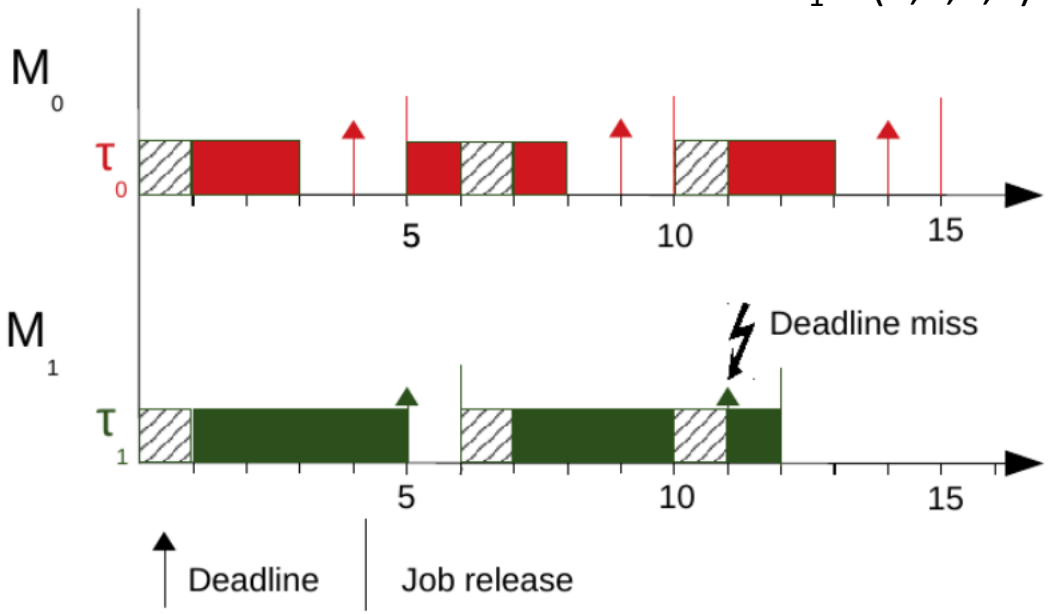
$$dbf''_{\tau_{M_k}}(t) \leq dfb'_{\tau_{M_k}}(t)$$

less pessimistic

Schedulability analysis

- Counterexample

τ_0 = (2,4,5,1)
τ_1 = (4,5,6,1)



The first busy period is not the worst case

Schedulability condition

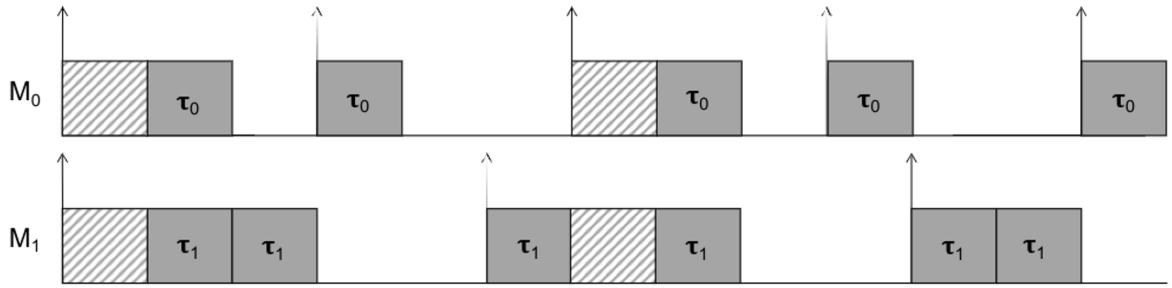
$$dbf''_{\tau_{M_k}}(t_1, t_2) \leq t_2 - t_1 \quad 0 \leq t_1 < t_2 \leq H$$

Task allocation

- Allocating task to cores is key to reduce the interference
- Well-known bin packing algorithms:
  - FFDU → First Fit Decreasing Utilization
  - WFDU → most widely used as it balances the load between cores
  - BF, NF, etc
- We need an allocation strategy that takes into account interference parameter I_i

Wmin allocator

- W matrix (n x n x H): takes into account the posible interference produced
- W is a binary matrix:
  - W_{ij} = 1 → τ_i provoques interference to τ_j
  - W_{ij} = 0 → otherwise
- τ_i and τ_j in the same core ➔W_{ij},W_{ji}=0
- τ_i is not broadcasting ➔ W_{ij},W_{ji}=0



t	W	Comment
0	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$	τ_0 and τ_1 release
1	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$	
2	$\begin{pmatrix}0 & 0\\1 & 0\end{pmatrix}$	τ_0 finishes its first activation
3	$\begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}$	τ_1 finishes its first activation
4	$\begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}$	
5	$\begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}$	τ_1 releases, but τ_0 is not active
6	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$	τ_0 releases while τ_1 is active
7	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$	
8	$\begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}$	τ_0 and τ_1 finish
9->15	$\begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}$	τ_0 and τ_1 are not active at the same time

Wmin allocates tasks to cores so W matrix is minimised

minimise $$maxW = \sum_{\forall k} maxW_k = \sum_{k=1}^{m} \sum_{\substack{\tau_i \in M_k \\ I_i \neq 0}} \sum_{\tau_j \notin M_k} I_j$$

s.t.

$$\sum_{\forall k} O_{ik} = 1 \qquad \forall i$$

$$\sum_{i \in k} U_i \cdot O_{ik} = U_{M_k} \qquad \forall k$$

$$U_{M_k} \leq 1 \qquad \forall k$$

$$\sum_{\substack{\tau_i \in M_k \\ I_i \neq 0}} \sum_{\tau_j \notin M_k} I_j = maxW_k \qquad \forall k$$

$$O_{ik} \in \{0, 1\}$$

$$U_{M_k}, maxW_k \geq 0$$

SETS AND INDICES

i	Tasks $\tau_i \in \{0, 1, 2, ..., n-1\}$
k	Cores $M_k \in \{0, 1, 2, ..., m-1\}$

PARAMETERS

C_i	Worst case execution time of τ_i
T_i	Period of τ_i
U_i	Theoretical utilisation of τ_i
I_i	Interference factor of τ_i over other tasks

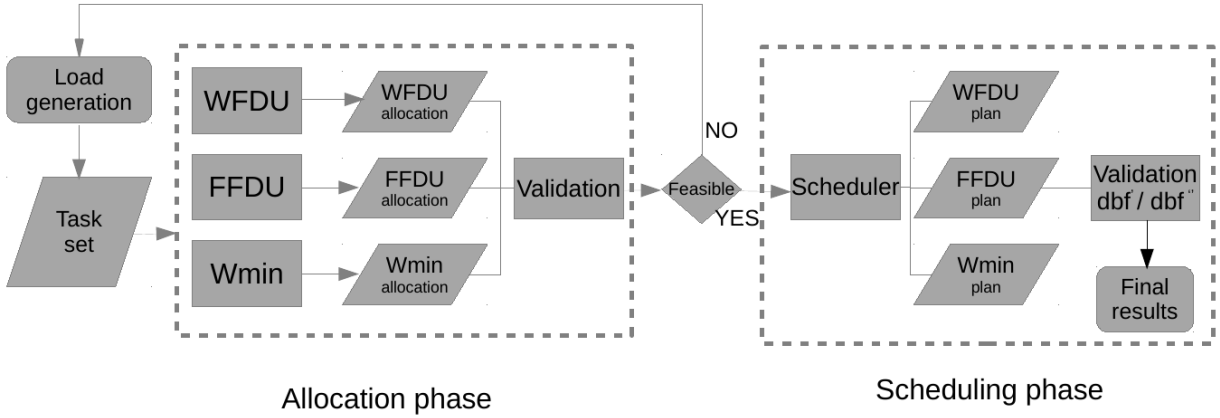
DECISION VARIABLES

O_{ik}	Allocation matrix. 1 if τ_i is allocated in core k and 0 otherwise.
U_{M_k}	Theoretical utilisation of core k.
$maxWk$	Maximum value of the sum of all elements of W for core k.
$maxW$	Maximum value of the sum of all elements of W for all cores.

Evaluation

To evaluate the proposal we created a simulation scenario divided into 5 elementary steps:

1. Load generation
2. Allocation phase
3. Allocation validation
4. Scheduling
5. Scheduling validation



Intel Core i7 16GB RAM, Gurobi optimizer 9.5

Evaluation

Number of cores	utilisation	Tasks	Broadcasting tasks	Interference over WCET (%)	Scenario
2 cores	1.1	4	2	10	1
				20	2
				30	3
	1.5			10	4
				20	5
				30	6
4 cores	2.1	12	3	10	7
				20	8
				30	9
	3			10	10
				20	11
				30	12
8 cores	4.1	20	5	10	13
				20	14
				30	15
	6			10	16
				20	17
				30	18
10 cores	5.1	28	7	10	19
				20	20
				30	21
	7.5			10	22
				20	23
				30	24

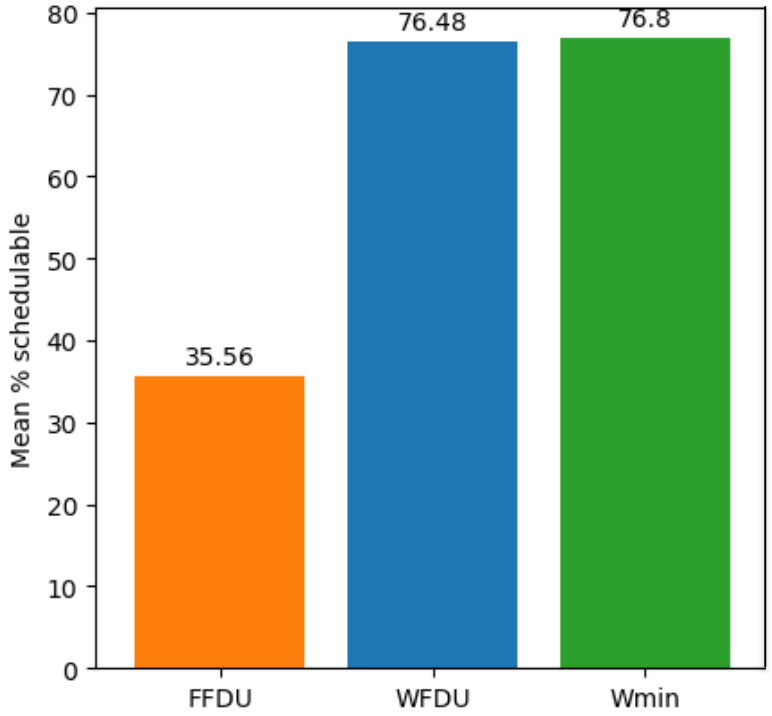
Evaluation

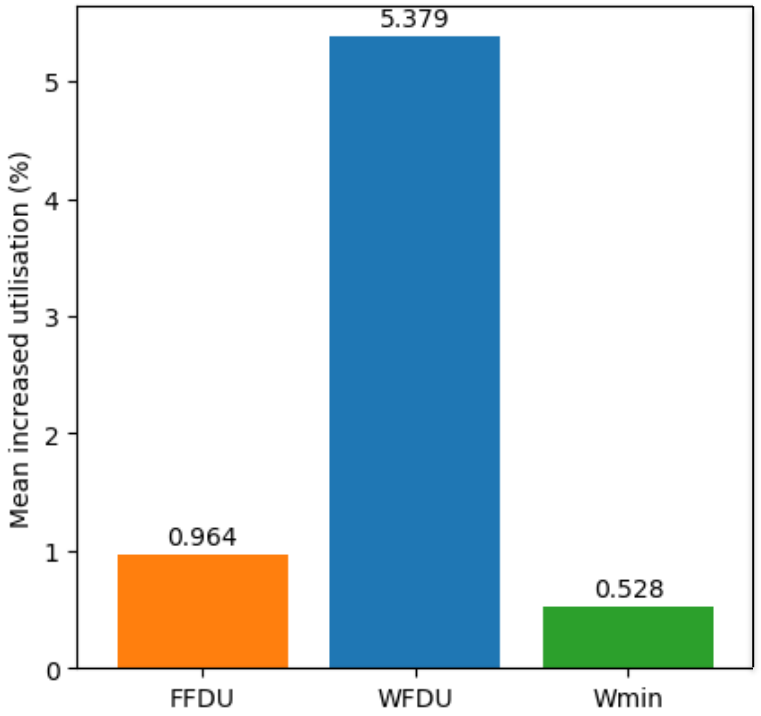
- Evaluated parameters
  - Allocators comparison:
    - Increased utilization: The increase in utilization with respect to the theoretical utilization
    - Schedulability ratio: Percentage of feasible plans over total task set with valid allocations (step5 vs step3)
  - Schedulability comparison
    - α': difference between real and estimated utilisation by dbf'
    - α'': difference between real and estimated utilisation by dbf''

Evaluation

Increased utilization and schedulability ratio

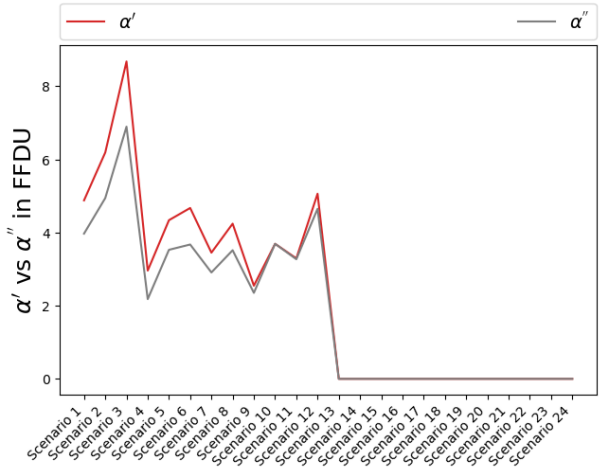
- WFDU has good schedulability but much increased utilisation.
- FFDU has the opposite behaviour
- Wmin presents the advantages of WFDU and FFDU.

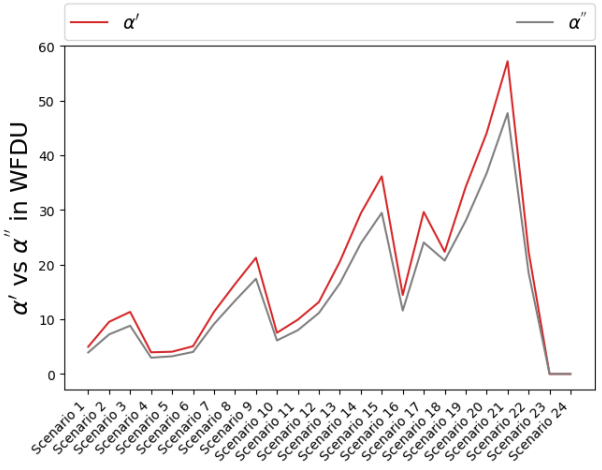


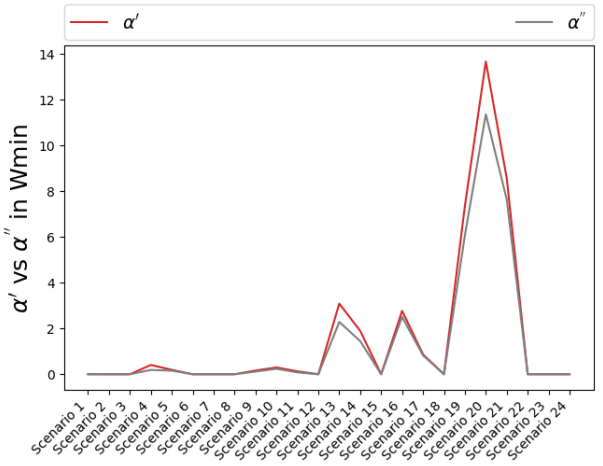


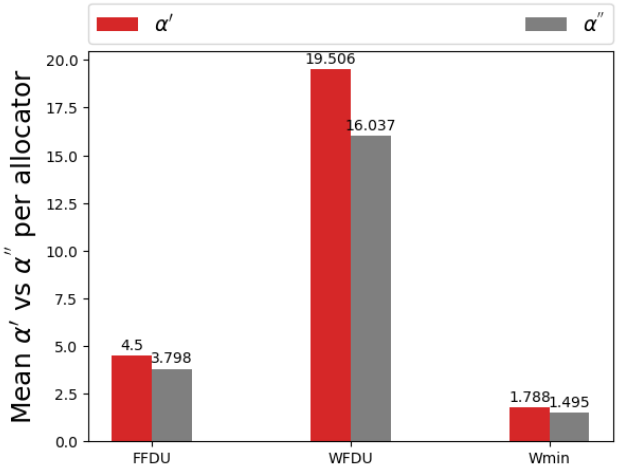
Evaluation

- α' and α''









Conclusions

- This talk is the result of the work published in:
  - José María Aceituno, Ana Guasque, Patricia Balbastre, José E. Simó, Alfons Crespo. Hardware resources contention-aware scheduling of hard real-time multiprocessor systems. J. Syst. Archit. Vol. 118 (2021)
  - Ana Guasque, José María Aceituno, Patricia Balbastre, José Simó, Alfons Crespo. Schedulability analysis of dynamic priority real-time systems with contention. The Journal of Supercomputing (2022)
- The model is implemented in Xoncrete tool
  - Eyesat 3U Cubesat
- Future work
  - Improve I_i calculation with more specific information
  - Improve schedulability analysis
  - Extend to other models