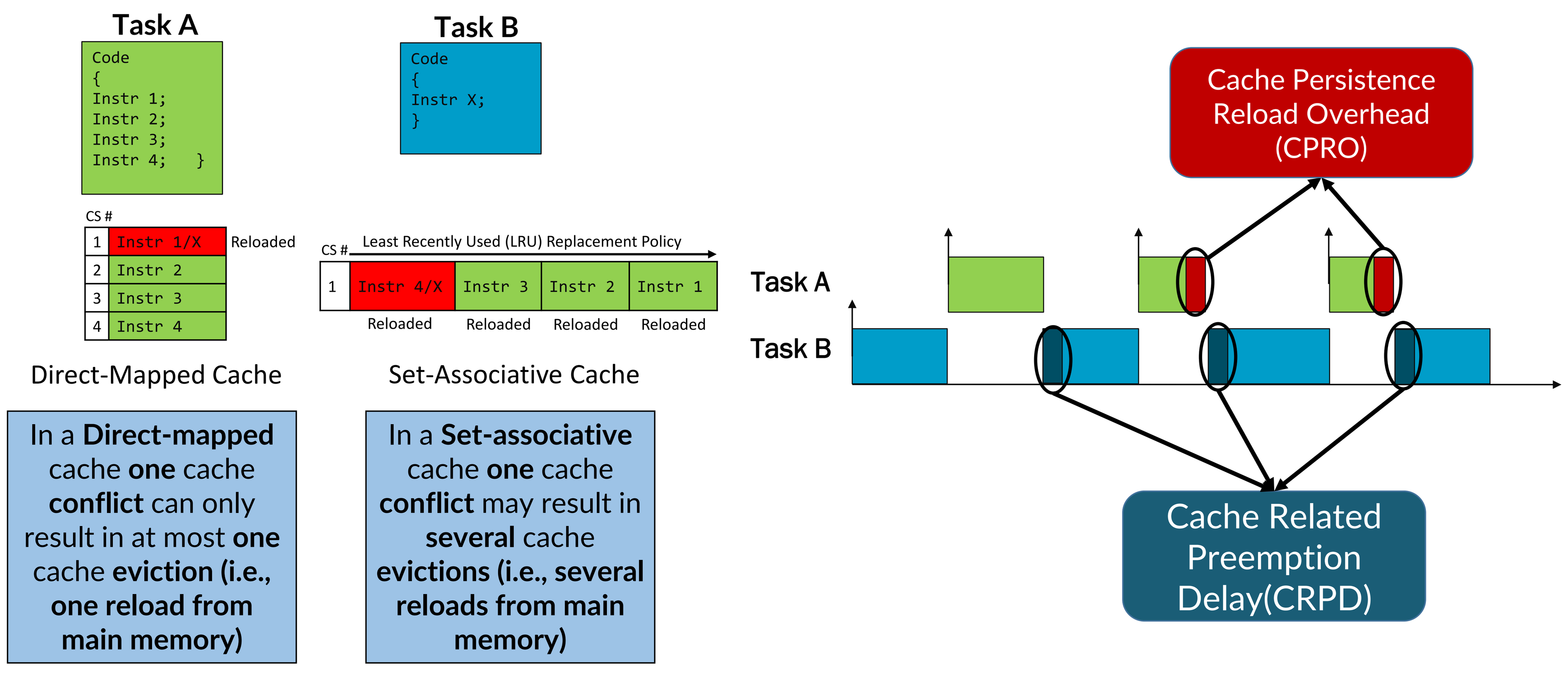
ResilienceP Analysis: Bounding Cache Persistence Reload Overhead for Set-Associative Caches

CISTER – Research Centre in Real-Time & Embedded Computing Systems

Syed Aftab Rashid, Geoffrey Nelissen, Eduardo Tovar
{syara;gnn;emt}@isep.ipp.pt

1. Motivation

- Caches induce **time variability** in task executions due to **CRPD/CPRO.**
- **Lower** priority tasks may need to account for **cache evictions** due to **preemptions** by the **higher** priority tasks **(CRPD).**
- **Memory demand** of the **preempting** tasks **depends** on the execution of **all** other tasks **(CPRO).**
- **SoA approaches** for **CPRO** calculation **only** consider **direct-mapped** caches.

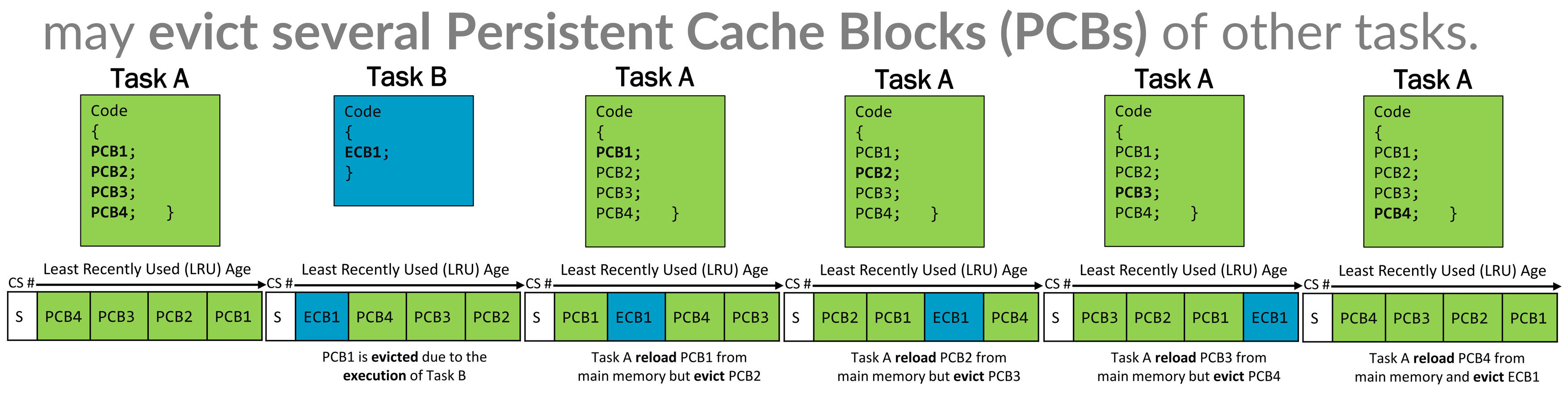


2. Contributions

- Three approaches to calculate CPRO for set-associative caches.
  - PCB-ECB approach
  - ResilienceP Analysis
  - Improved ResilienceP Analysis
- The proposed approaches accounts for both CRPD and CPRO and dominates the SoA Resilience analysis that only accounts for CRPD.

3. PCB-ECB Approach

- In a **set-associative** cache **one Evicting Cache Block (ECB)** of a task may **evict several Persistent Cache Blocks (PCBs)** of other tasks.



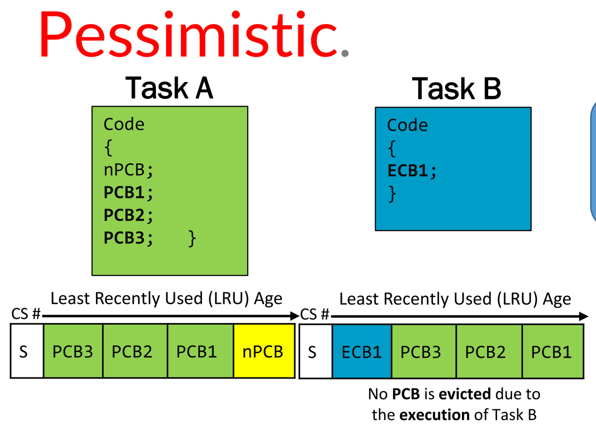
If **one** or more **ECBs** of **any** task (other than the **task under analysis**) are **mapped** to cache set **S** then **all PCBs** (of the **task under analysis**) in **S** may be **evicted** and hence must be acc**ounted** for in the **CPRO**.

$$\rho^{set,s}_{j,i} = d_{mem} \times \min\left(CPRO^s_j, CPRO^s_{hep(i)\setminus\tau_j}\right)$$

$$CPRO^s_j = |PCB^s_j| \qquad CPRO^s_{hep(i)\setminus\tau_j} = \begin{cases} k & \text{if } \bigcup_{\forall\tau_k\in hep(i)\setminus\tau_j} ECB^s_k \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

4. ResilienceP Analysis

- Assuming **one ECB** of **any** task (other than the **task under analysis**) in cache set **S** may evict **all PCBs** (of the **task under analysis**) in **S** is Pessimistic.



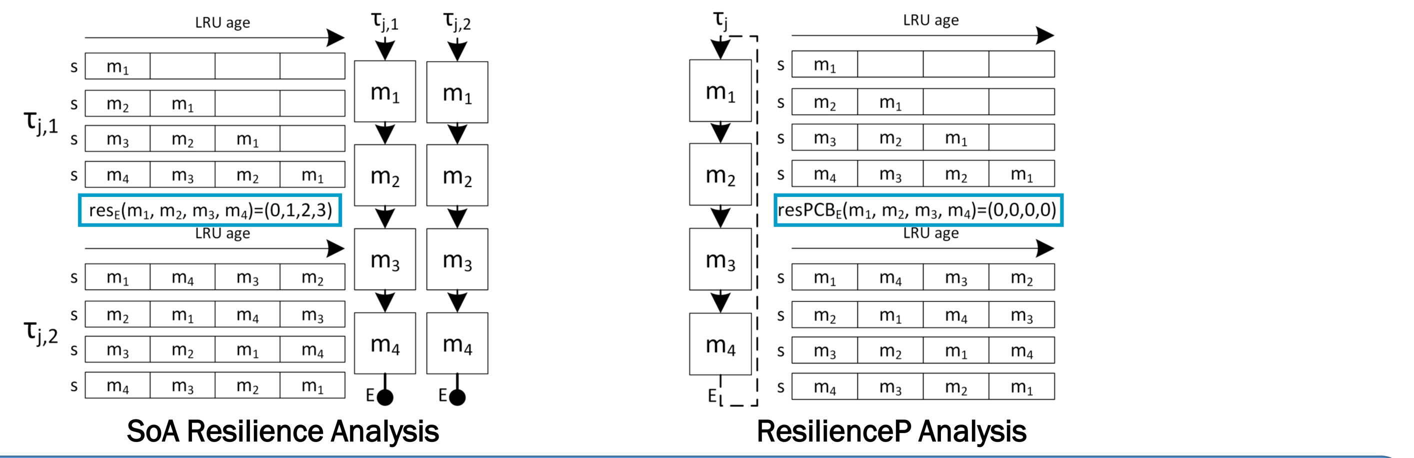
CPRO using PCB-ECB Approach = 3
Actual CPRO = 0

The Actual **CPRO** depends on the **Resilience** of **PCBs**

$$Resilience(PCB) = (Cahe\ Associativity - 1) - Maximum\_Age(PCB)$$

4.1 SoA Resilience Analysis underestimate the CPRO

- **SoA Resilience Analysis soundly** estimates the **maximum-age** (and Resilience) of **UCBs** but may **overestimate** the **Resilience** of **PCBs.**

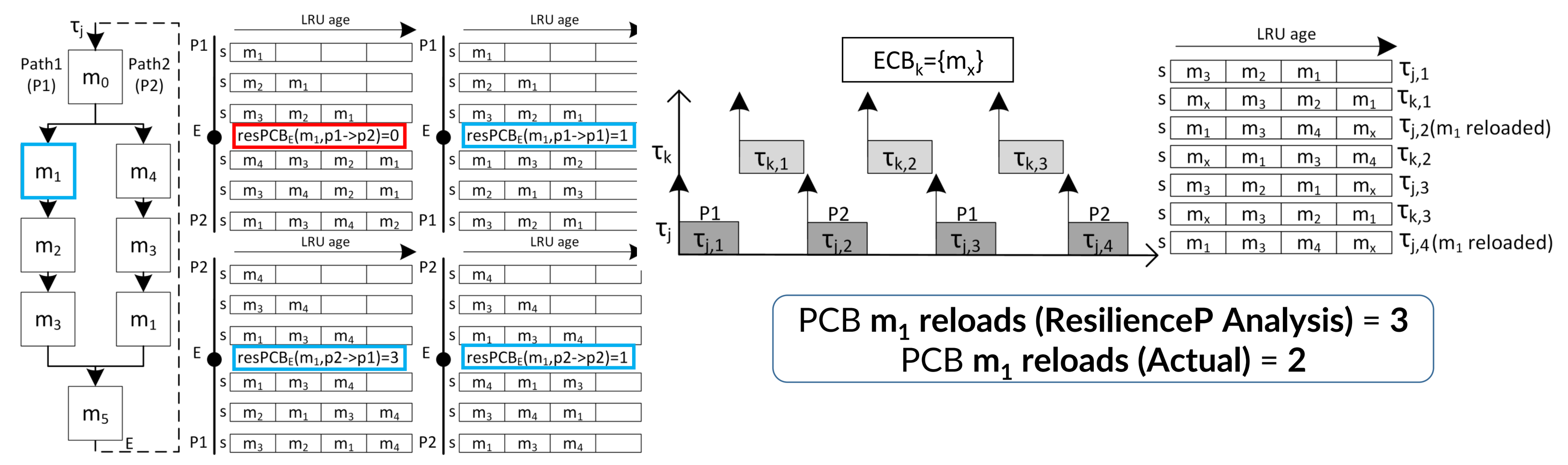


The **ResilienceP analysis** accounts for the **overestimated** resilience of **PCBs** in the **SoA Resilience analysis** by calculating the **maximum-age** of **PCBs** assuming the task is **cyclic**.

$$\rho^{res,s}_{j,i} = d_{mem} \times \left| PCB^s_j \setminus \left\{ m_j | res_{PCB}(m_j) \geq \sum_{\forall\tau_k\in hep(i)\setminus\tau_j} |ECB^s_k| \right\} \right|$$

5. Improved ResilienceP Analysis

- For **tasks** with **more** than one **execution paths**, **resilience** of PCBs may **vary** depending on the **execution path taken** by the task.



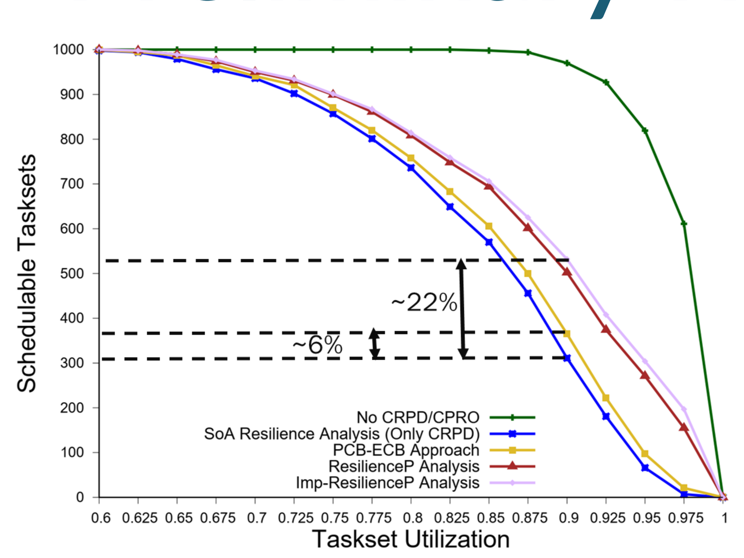
Always considering the **minimum** resilience of PCBs for **all** jobs of a task executing in a **time interval t** may lead to **overestimating** the **total CPRO** the task may **suffer** in **t**.

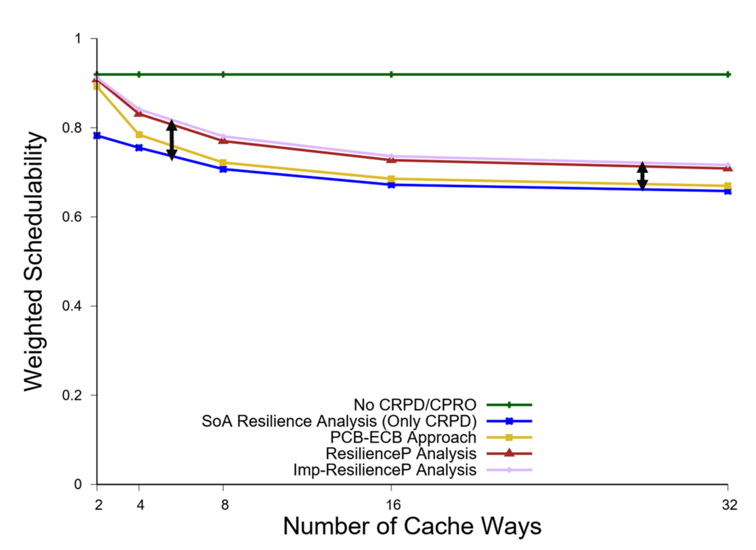
By considering the **variation** in the **resilience** of **PCBs** over different **jobs** of a task **executing** in a **time interval t,** the **improved** resilienceP analysis results in much **tighter CPRO bounds** than the **ResilienceP** analysis.

Disturbance (D)	Number of jobs of τ_j (J): 2	3	...	$\lceil \frac{t}{T_j} \rceil$
1	$\min(1,x)$	$\min(2,x)$	...	$\min(\lceil \frac{t}{T_j} \rceil - 1, x)$
2	$\min(1,x)$	$\min(2,x)$	...	$\min(\lceil \frac{t}{T_j} \rceil - 1, x)$
...	...	...	...	...
$\geq k$	1	2	...	$\lceil \frac{t}{T_j} \rceil - 1$

CPRO Table for PCBs

6. Preliminary Results





7. Future Work

- In future, We will **investigate** how to **efficiently** build the **CPRO table** for **PCBs** under the **improved ResilienceP** analysis.
- We also plan to **extend** the analysis to **cache hierarchy** and **shared caches** in **multi-core** systems.
- Perform **extensive** experiments to **evaluate** our solutions

Co-financed by Unidade de I&D CISTER - CEC/04234

CISTER Research Centre
ISEP, Polytechnic Institute of Porto
Rua Dr. Antº Bernardino de Almeida, 431
4249-015 Porto, Portugal

+351 228 340 502
www.cister-labs.pt
info@cister-labs.pt
facebook.com/cisterlabspt