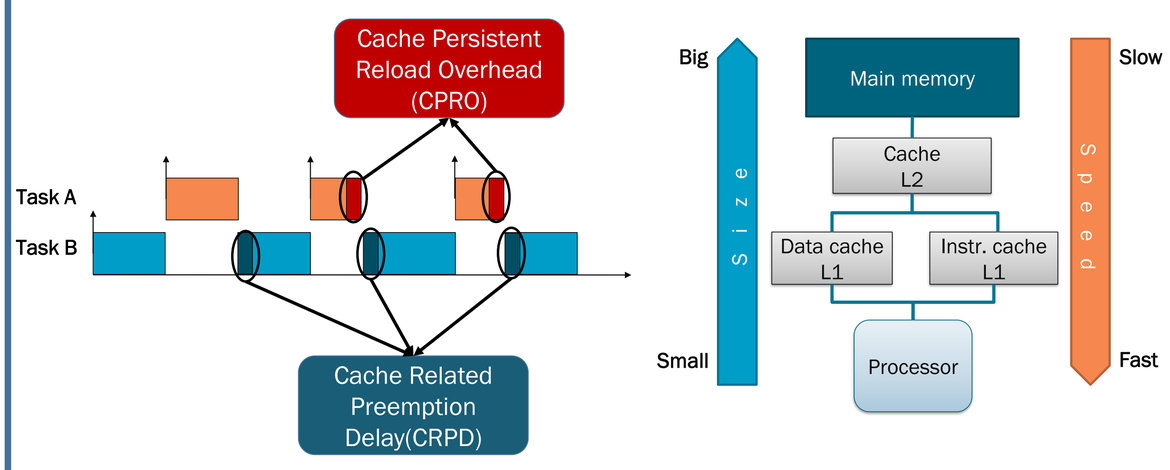
Integrating the Calculation of Preemption and Persistence Related Cache Overhead

CISTER - Research Center in
Real-Time & Embedded Computing Systems

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

1. Motivation

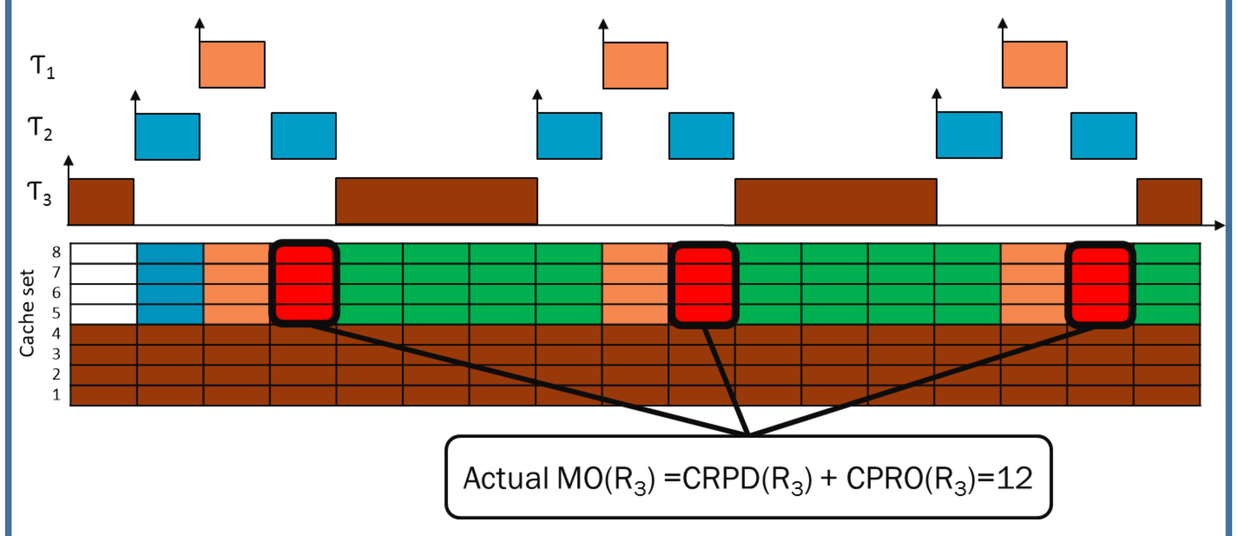
- Caches produce **time variability** in task **WCET/WCRT**
- Low priority tasks may need to account for **cache evictions** due to **preemptions** by the high priority tasks **(CRPD).**
- **Memory demand** of the **preempting** tasks depends on the execution of **all** other tasks **(CPRO).**
- **Independent** calculation of **CPRD** and **CPRO** may lead to **overestimations** in **WCET/WCRT** of tasks.



2. Contributions

- Mutual **dependency** between **CRPD** and **CPRO** may lead to **double** accounting of **same** cache block evictions
- We **identify** the cache blocks whose evictions can be accounted **twice**.
- We **integrate** the calculation of CRPD and CPRO to **remove** the **pessimism** in the existing analysis
- The **improved** analysis ensures that the **same** cache block evictions are accounted only **once** either in CRPD or CPRO.

3. Example



SoA Memory Overhead (MO) Calculation

- Using **UCB-union** Approach, **CRPD(R_3)**= 3 x ($CRPD_{3,1}$ + $CRPD_{3,2}$)=**12**
- Using **CPRO-Union** Approach, **CPRO(R_3)**= 2 x ($CPRO_{2,3}$ + $CPRO_{1,3}$)=**8**
- **MO(R_3) = 12+8=20** ➡ Overestimation!

4. Existing Approaches for CRPD and CPRO Calculation

4.1 UCB-union Approach

- Accounts for the eviction of **Useful Cache Blocks (UCBs)** of the **preempted** task τ_i due to **preemptions** by the high priority task τ_j
- **UCB-union** approach considers that the **UCBs** of all **intermediate** priority tasks in **aff(i,j)** = **hp(i)** ∩ **lp(j),** can be evicted by τ_j
- **ECBs** of the preempting task τ_j upper **bound** the number of **UCBs** of all tasks in **aff(i,j)** it can **evict**

$$CRPD_{i,j} = \left| \left(\bigcup_{\forall k \in aff(i,j)} UCB_k \right) \cap (ECB_j) \right|$$

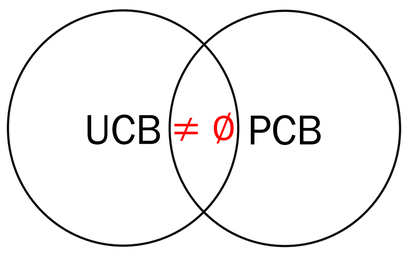
4.2 CPRO-Union Approach

- Accounts for the eviction of **Persistent Cache blocks (PCBs)** of the **preempting** task τ_j executing during the response time of τ_i
- **PCBs** of τ_j can be evicted due to executions of tasks in **hp(i).**
- CPRO-union approach considers that the **ECBs** of all tasks in **hp(i)** can evict the **PCBs** of τ_j.

$$CPRO_{j,i} = \left| PCB_j \cap \left(\bigcup_{\forall k \in hp(i) \setminus j} ECB_k \right) \right|$$

5. Proposed Solution

5.1 Identifying Double Accounting Cache Blocks



when a cache block is both UCB and PCB its eviction may be counted twice both in CRPD and CPRO

5.2 CRPD-aware CPRO-Union Approach

- Only tasks in **hp(j)**, can contribute **both** to $CPRO_{j,i}$ and $CRPD_{i,j}$

$$CPRO_{j,i} = \left| PCB_j \cap \left(\bigcup_{\forall k \in hp(i) \setminus j} ECB_k \right) \right|$$

$$CPRO^{imp}_{j,i} = \left| PCB_j \cap \left(\left(\bigcup_{\forall k \in aff(i,j)} ECB_k \right) \cup \left(\bigcup_{\forall l \in hp(j)} ECB_l \setminus UCB_j \right) \right) \right|$$

Only contribute to CRPD, thus is the same as for the CPRO-Union approach

UCBs whose evictions are accounted for in the CRPD are not considered in CPRO

6. Future Work

- Presented approach is applicable **only** if the CRPDs are calculated **using** the **UCB-union** approach. In future works, we plan to **extend** the analysis to less pessimistic **multi-set** approaches used for CRPD and CPRO calculations
- Possibility of having a **single** term in the **WCRT analysis** that accounts for both CRPD and CPRO.
- Extensive **experimental** evaluation using available benchmarks by **varying** different system parameters.

References

[1] S. Altmeyer, R. Davis, C. Maiza et al., "Cache related pre-emption delay aware response time analysis for fixed priority pre-emptive systems," in RTSS'11. IEEE, 2011, pp. 261–271
[2] S. A. Rashid, G. Nelissen, D. Hardy, B. Akesson, I. Puaut, and E. Tovar, "Cache-persistence-aware response time analysis for fixed-priority preemptive systems," in 2016 28th Euromicro Conference on Real-Time Systems. IEEE, 2016, pp. 262–272.

EMC2 project ARTEMIS/0001/2013 - JU grant nr. 621429. Co-financed by:
COMPETE 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundos Europeus Estruturais e de Investimento FCT Fundação para a Ciência e a Tecnologia

CISTER Research Centre/INESC-TEC
ISEP, Polytechnic Institute of Porto
Rua Dr. Antº Bernardino de Almeida, 431
4249-015 Porto, Portugal
+351 228 340 502
www.cister.isep.ipp.pt
cister-info@isep.ipp.pt
facebook.com/cisterrealtime
isep Instituto Superior de Engenharia do Porto
INESCTEC