Towards Timing Analysis of Multi-core Platforms for Hard Real-Time Systems

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

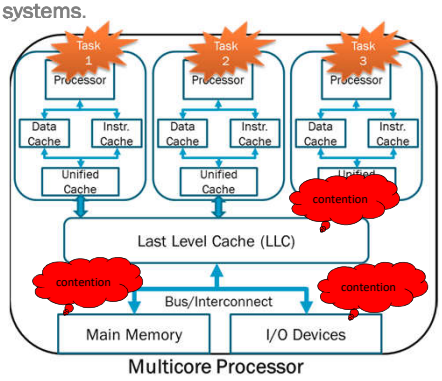
U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Cyber-Physical Systems Week

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

1. Motivation

- Commercially available Off-The-Shelf (COTS) **multi-core processors (MCPs)** have become **main** stream.
- Several **advantages** over **single-core/custom** build hardware.
- Design mantra of **"average case faster"** makes **MCPs** a **popular** choice in **low-criticality/soft** real-time systems.
- The **performance oriented** design of **MCPs** makes them **non-deterministic**, e.g. **multiple-cores, pipelines, multi-level cache**.
- The **non-determinism** of MCPs **restricts** their use in **hard real-time systems**.



WCET of a task executing on **one** core of a **MCP** not only depends on the task itself but also on the behavior of **tasks** executing in **parallel** on the **other** cores.

The **intra-** and **inter-core interference** due to contention for **shared resources** between concurrently executing task must be **bounded** in order to provide **deterministic** bounds on the **WCET** and **WCRT** of tasks running on a **MCP**

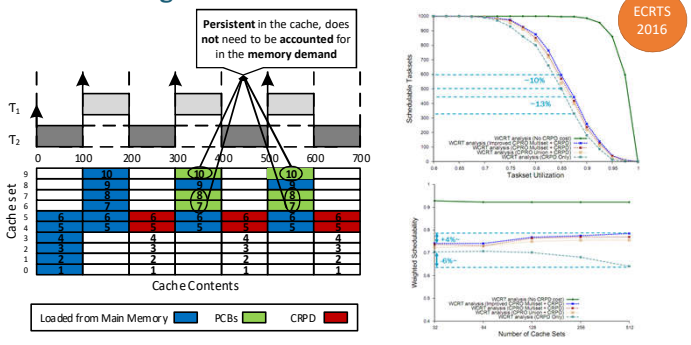
2. Objectives

- We intend to provide **solutions** that can be used to **quantify** and analyze the **non-determinism** arising from the **sharing** of **two** main **resources** in MCPs, i.e., **caches** and **interconnects**.
- Accurately **quantify** the **cache** related contention in **single core** platforms.
- Bounding the interference due to **cache hierarchy** and **last-level shared cache (LLC)** in multicore platforms.
- Model the **inter-core interference** due to the sharing of **Bus/interconnects** in a MCP.
- Develop a new **timing analysis** taking into account the interference caused by **both caches** and **interconnects** and their **impact** on the **timing** properties of **tasks** running on MCPs

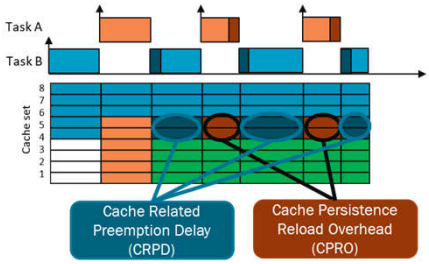
3. Completed Work

3.1 Introducing the Notion of Cache Persistence

ECRTS 2016



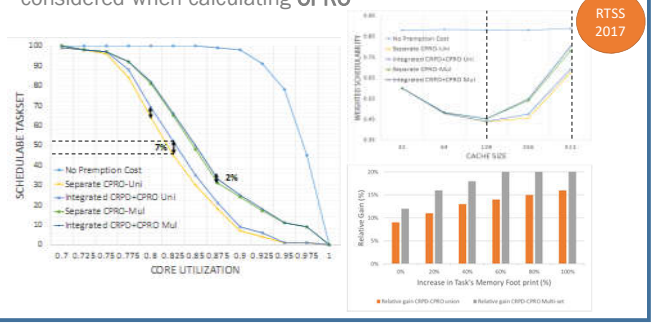
3.2 Integrated Analysis of Cache Related Preemption Delay (CRPD) and Cache Persistence Reload Overhead (CPRO)



Independent calculation of **CRPD** and **CPRO** may result in **over-estimating** the actual **memory overhead** suffered by tasks due to **preemptions**

- Memory **overheads** already accounted for in the **CRPD** will **not** be considered when calculating **CPRO**

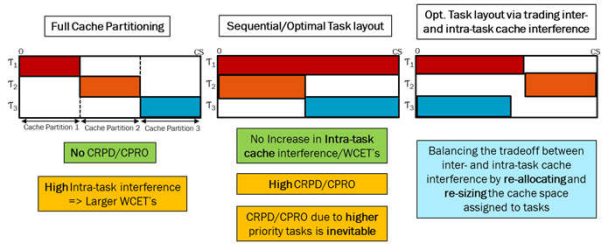
RTSS 2017



4. Ongoing Work

4.1 Optimize task layout in memory to reduce CRPD/CPRO and improve schedulability

- Task set $\{T_1,T_2,T_3\}$ with cache requirements {CS,CS/2,CS/2}



- Use **cache coloring** to assign cache space to tasks.
- **Optimize** cache **color assignment** of tasks by using an optimization **algorithm**, e.g., **Simulated Annealing**.

6. Future Work

- Extending the **cache persistence** analysis to cache **hierarchy** and **last-level shared cache (LLC)** in multicore platforms.
- To **bound** the interference generated by **interconnects** used in MCPs by assuming a more **predictable** task execution **model** e.g., **PREM**, **3-phase/AER** task model
- **Combined** analysis of the interference caused by **both** caches and **interconnects** considering the **cascading** effect between them.

Supported by National Funds through FCT (Portuguese Foundation for Science and Technology) within the CISTER Research Unit (CEC/04234);

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