Timing Properties of the LEON3-based GR712RC Board. Implications on Task Scheduling

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Francisco J. Cazorla, Mikel Fernandez
Jaume Abella, Eduardo Quiñones
Gabriel Fenandez

Marco Zulianello (TO)
Luca Fossati
Christophe Honvault

Francisco J. Cazorla
PhD. Researcher and Director of the CAOS group at BSC
(www.bsc.es/caos) (francisco.cazorla@bsc.es)

Software Systems Division & Data Systems Division
Final Presentation Days
11-12th December 2013, ESTEC

BSC

- Barcelona Supercomputing Center
- Spanish national research center (www.bsc.es)
  - +400 people at the end of 2012 (>80% are researchers)
- Areas of research:
  - Life Sciences
  - Earth Sciences
  - Computer Applications
  - Computer Sciences. It comprises several research groups
    - Compiler group
    - Programming Models group
    - …

The CAOS group

- Computer Architecture/Operating System (CAOS) (www.bsc.es/caos)
  - 16 people (4 PhDs, 2 Engineers, 10 PhD students)
  - Research lines and collaborations:
    - HPC systems: IBM; Networking systems: Sun Microsystems
    - Real-Time systems: European Projects
      - FP7 MERASA, parMERASA, P-SOCRATES
      - FP7 PROARTIS, PROXIMA (coordinated projects)
      - VeTeSS ARTEMIS Project
    - Projects with ESA
      - 2 finalized, 1 running

Agenda

- Why multicores in the space domain?
- Inter-task interferences in the NGMP and the GR712RC
- Effect on task scheduling
- Conclusions and Future work

Agenda

- **Why multicores in the space domain?**
- Inter-task interferences in the NGMP and the GR712RC
- Effect on task scheduling
- Conclusions and Future work

Introduction

- Critical Real-time Embedded Systems (CRTESs)



- Some of the main requirements of hard real-time systems
  - Functional correctness
  - **Timing correctness**
- Provide higher functional value to keep competitive edge
  - Increasing portion of the system value provided by the SW

Requirements

- Value-added safety-related functions are becoming more complex
- More computational power required to run them in a timely manner
  - Average performance → guaranteed performance
- Multi-core processors can theoretically provide the required performance!

Multi-cores for CRTES

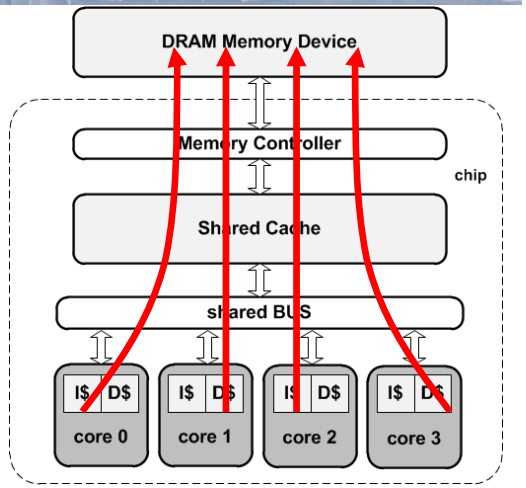
- Pros:
  - Better performance per watt than and simpler than single-core processors
  - Enable co-hosting mixed-criticality applications
    - Hardware utilization is maximized, while SWaP costs are reduced.
- Cons:
  - Require functional isolation
    - Low-criticality applications must not affect high-criticality ones [1]
  - Harder to provide WCET estimates
    - Because of inter-task interferences!

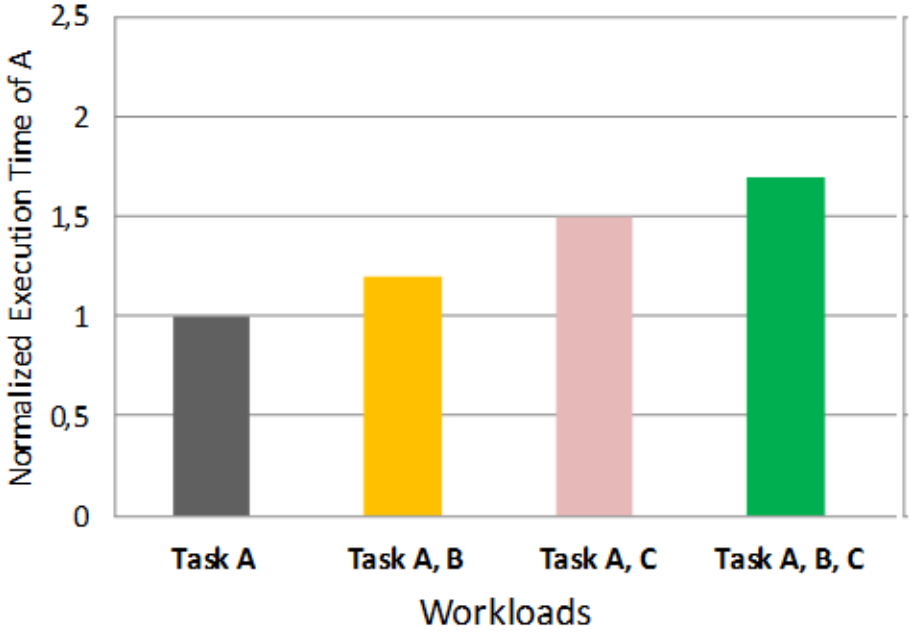
[1] ESA contract 4200023100, System Impact of Distributed Multi-core Systems

Inter-task interferences

- Appear when several tasks that share a hardware resource want to access to it simultaneously

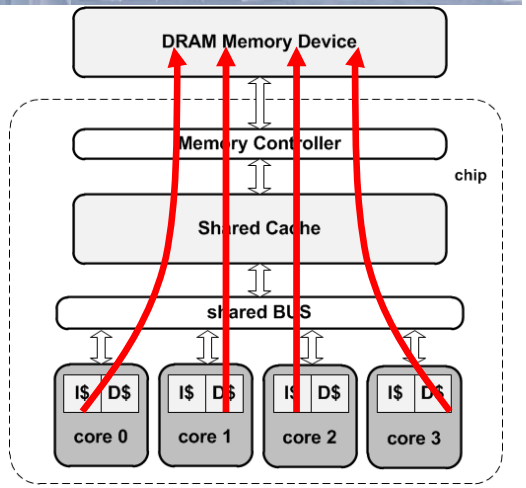
- The Execution time, and hence the WCET, of a task in a multi-core depends on the co-running tasks!

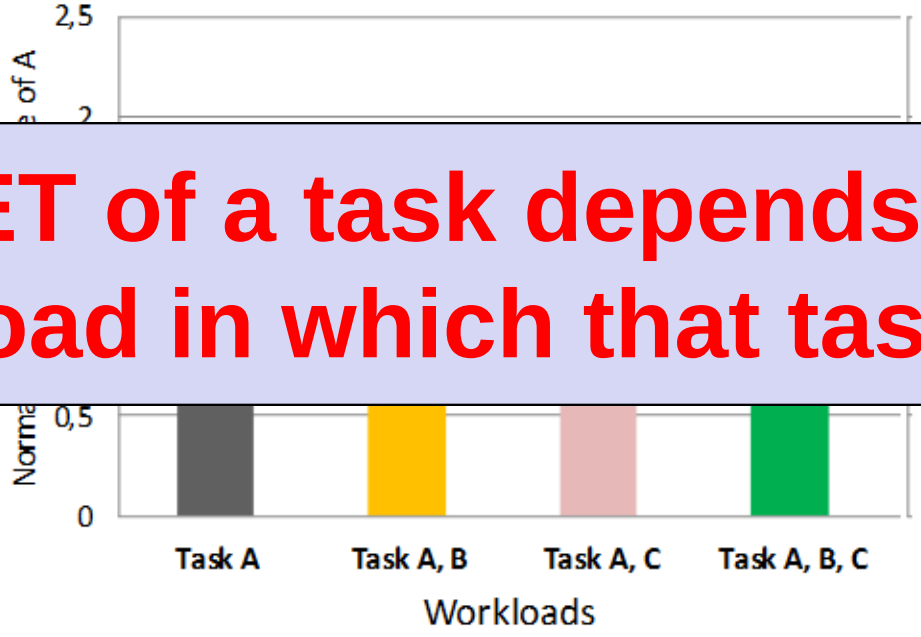




Inter-task interferences

- Appear when several tasks that share a hardware resource want to access to it simultaneously
- The Execution time, and hence the WCET, of a task in a multi-core depends on the co-running tasks!





WCET of a task depends on the workload in which that task runs!!!

State of the art (hardware proposals)

- Several proposals developed to ease the computation of WCET estimates for multicores (MERASA, ACROSS, GENESYS, PRET, TTA, PROATIS, PREDATOR ...)
  - Either by means of removing interactions between tasks or
  - upper-bounding the maximum interaction between tasks
  - NPI activity between BSC and ESA.
    - Title: Architectural solutions for the timing predictability of next-generation multi-core processors
    - Objective: Creating hardware support for taking inter-task interferences into account when computing WCET estimations for the NGMP (simulator)
    - People: Javier Jalle, Francisco J. Cazorla, Eduardo Quiñones, Luca Fossati, Marco Zulianello

State of the art (hardware proposals)

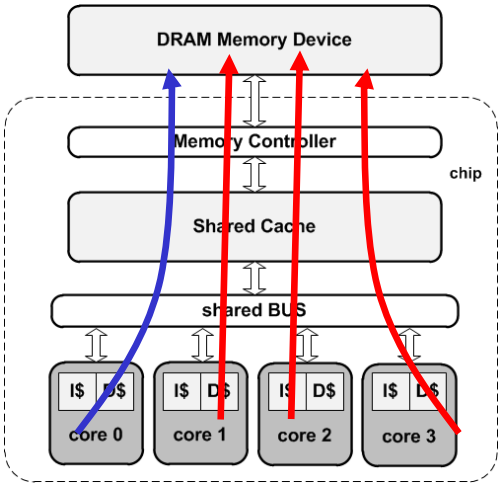
- Current multicores do not implement those hw features
  - It will take several years to be implemented
  - Industry cannot benefit nowadays from those proposals
  - **COTS multicore processors have to be used instead**

Agenda

- Why multicores in the space domain?
- **Inter-task interferences in the NGMP and the GR712RC**
- Effect on task scheduling
- Conclusions and Future work

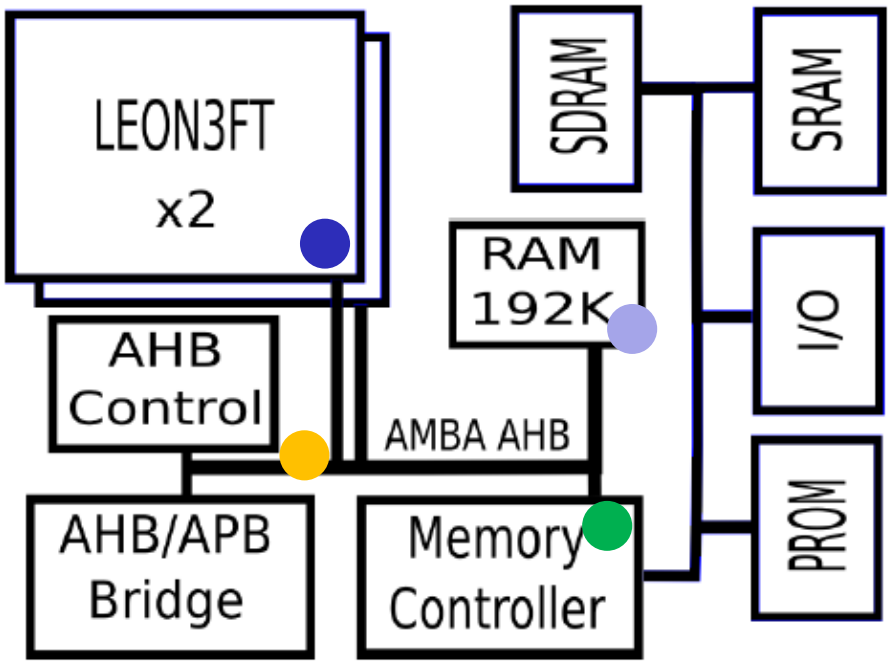
Objectives of the activity

- Provide a methodology to measure the real-time capabilities of multi-core architectures and, in particular, of the NGMP.
  - As part of this activity we designed a benchmark suite
    - Suitable to exercise the new NGMP multicore processor
    - Capable of generating different inter-task interference scenarios that may arise in the NGMP processor
- Impact on timing analysis of inter-task interferences



App RSK_1 RSK_2 RSK_3

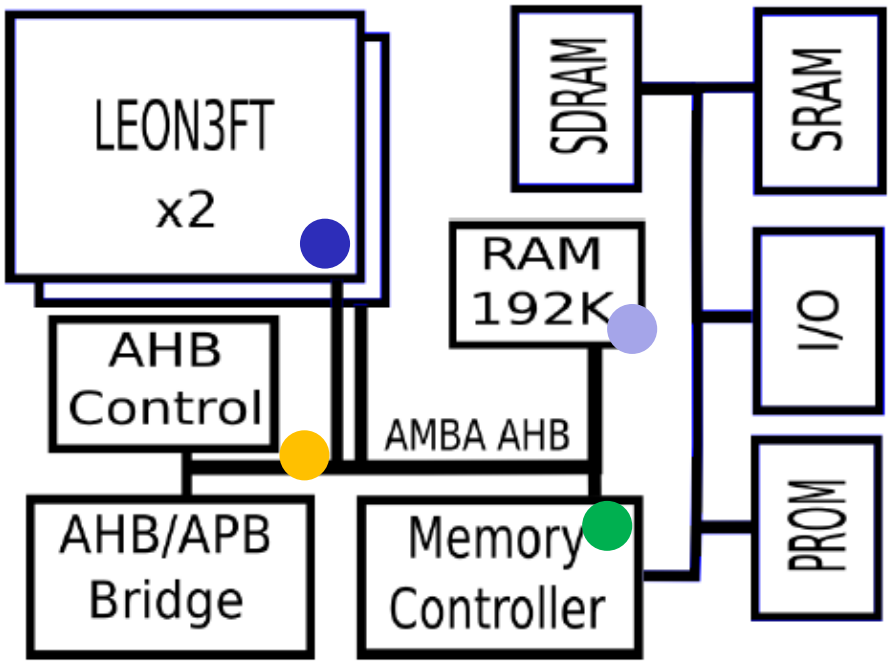
The boards: ML510 and GR712RC



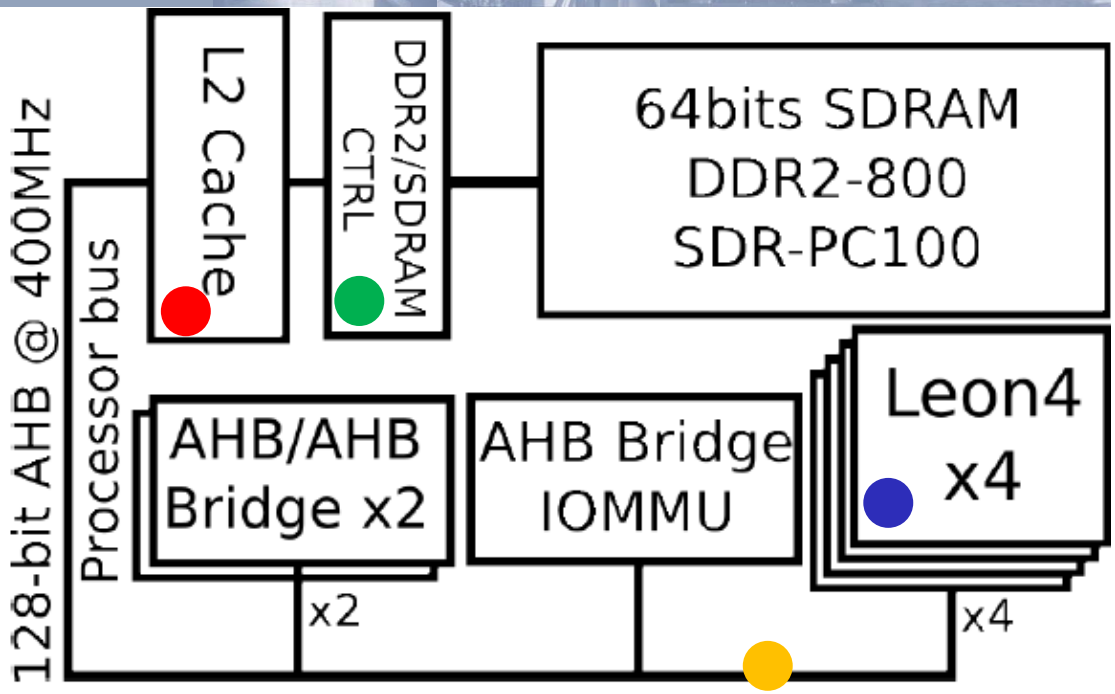
(a) GR712RC (LEON3)

- 2 LEON3 processors
- Per-core 16KB Data and I. caches
- Cores are connected to the on-chi SRAM and the memory controller through an AMBA AHB bus

The boards: ML510 and GR712RC



(a) GR712RC (LEON3)



(b) ML510 (LEON4)

- 2 LEON3 processors
- Per-core 16KB Data and I. caches
- Cores are connected to the on-chip SRAM and the memory controller through an AMBA AHB bus

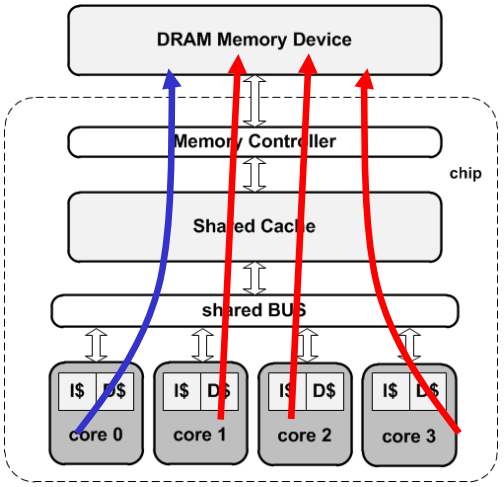
- 4 LEON4 cores
- Per-core 16KB Data and I. caches
- Shared bus to the L2, 256KB L2,
- Shared path to memory path

esa

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

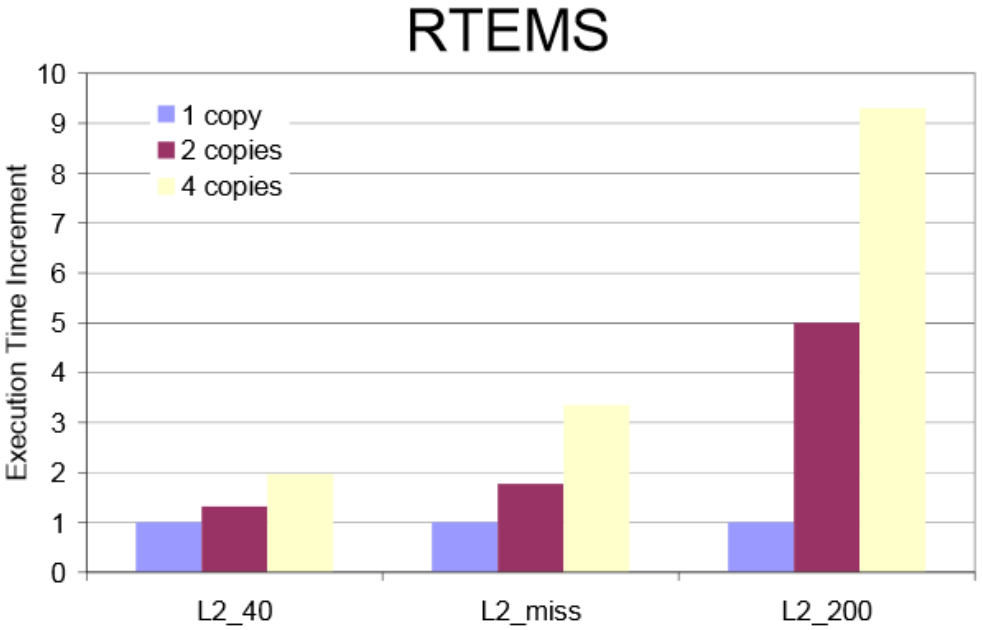
Microbenchmarks (Resource Stressing Benchmarks)

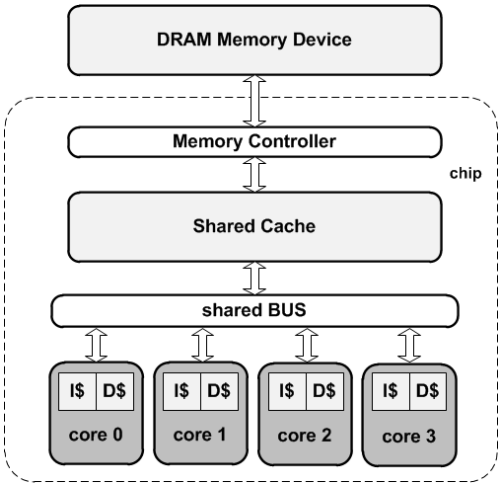
- Aka resource stressing benchmarks (RSK)
- Single-behavior kernels that constantly access a shared resource
  - Put high pressure on that resource
- Used as corunners to determine the slowdown a given application may suffer due to conflicts in that resource
- Examples of RSK
  - $L2_{40}$ (bus): traversal of a 40KB vector
    - 1 copy fits in L2, 4 copies fit in L2
  - $L2_{miss}$(bus,mem): traversal of a1MB vector
    - 1 copy misses in L2, 4 copies miss in L2
  - $L2_{200}$ (bus,mem, cache): traversal 200KB vector
    - 1 copy fits in L2, 4 copies miss in L2



App RSK_1 RSK_2 RSK_3

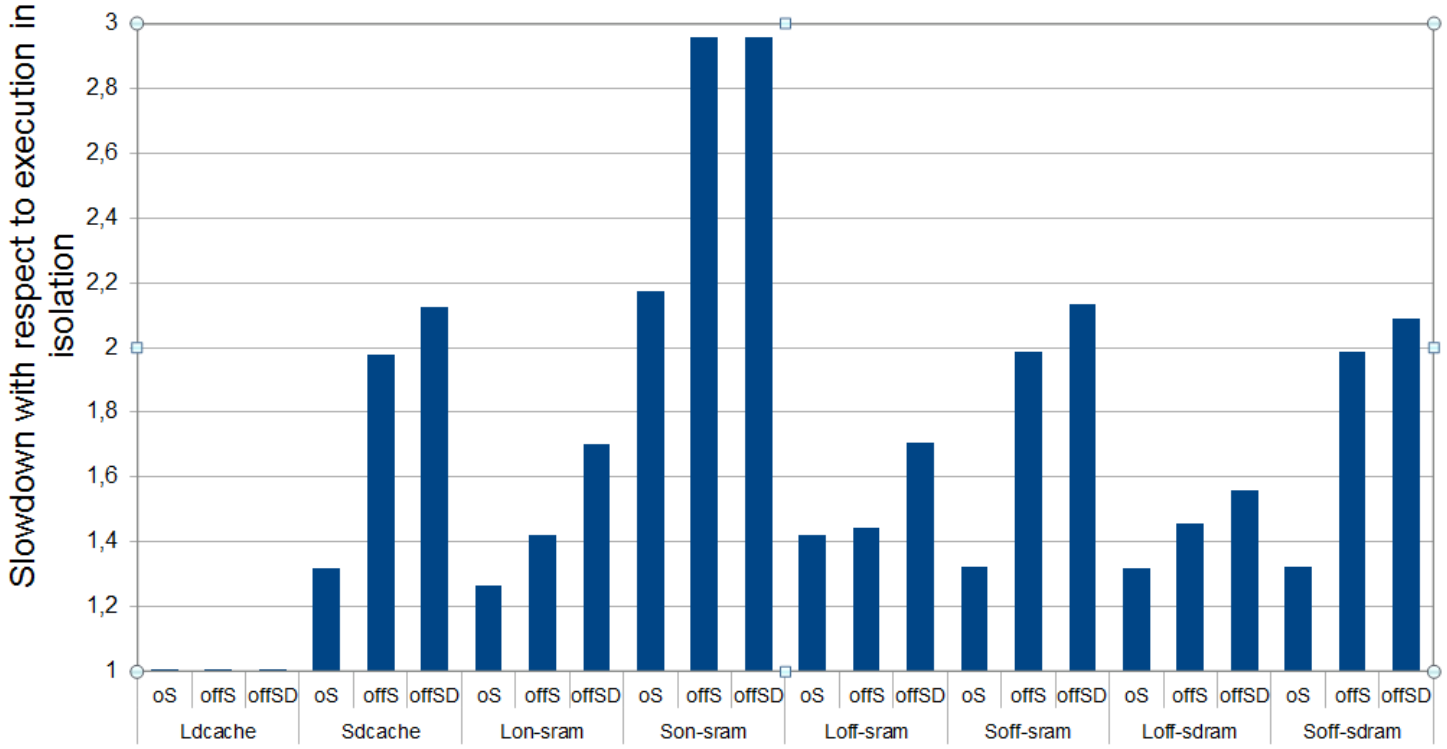
NGMP: Maximum slowdown a task may suffer

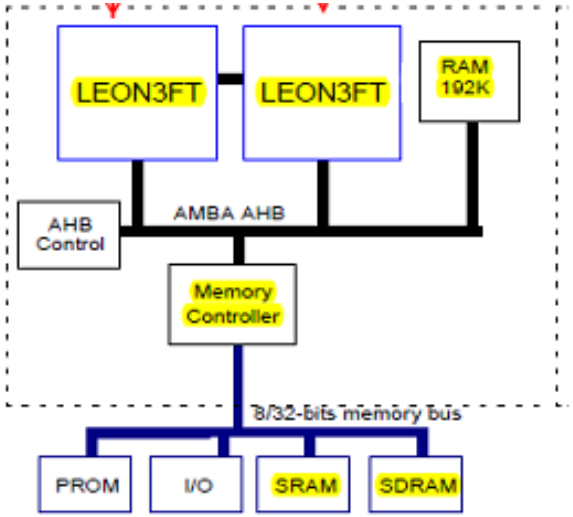




- Incurred slowdown (RTEMS)
  - $L2_{40}$ (bus): - 95%
  - $L2_{miss}$ (mem, bus): - 3.4x
  - $L2_{200}$ (mem, cache, bus): - 9x
- Suffered slowdown
  - $L2_{40\text{-stores}}$ (mem, cache, bus) - 20x

GR712RC: Maximum slowdown a task may suffer





- Tasks accessing on-chip RAM are the one affected the most (~3x)
- Tasks accessing the off-chips S(D)RAM less affected (~2.2x)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Conclusions

- Observations:
  - Inter-task interferences have a significant impact on tasks observed execution times on COTS multicores
  - NGMP observed slowdown due to inter-task interferences is higher than for the GR712RC
    - Higher number of cores
    - Inclusion of a shared LLC (L2)
- Challenges:
  - SW level: Impact on task allocation and scheduling
  - HW level: HW-support for inter-task interferences

Agenda

- Why multicores in the space domain?
- Inter-task interferences in the NGMP and the GR712RC
- **Effect on task scheduling**
- Conclusions and Future work

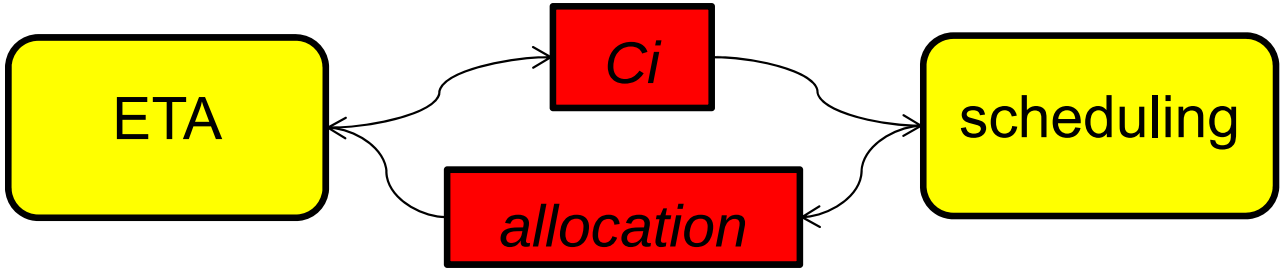
Timing Analysis in multicore COTS*

- Execution Time Analysis (ETA)
  - Deriving an Execution Time Bound bound (ETB) for each task
  - Computed assuming that the task runs in isolation
- Response Time Analysis (RTA)
  - Accounts for effects of system-level aspects such as
    - interrupts, blocking times when arbitrating access to software-level resources, and the preemptions that result from priority-driven scheduling
  - Derives a Response Time Bound (or RTB)
  - Serves as the compositional bridge between ETA and scheduling

* Work done in collaboration with Tullio Vardanega (University of Padua)

ETA-scheduler dependence in multicore

- Inter-task interferences affect task ETB in multicores
  - Accounting their effect in the RTA phase is impractical
  - Scheduling → determines the inter-task interferences a task suffers
- From scheduler standpoint
  - ETB estimations for each task (Ci) are first computed to determine a feasible schedule of the whole task set
  - Scheduling decisions affect the tasks' ETB
- From ETA standpoint
  - To determine ETB for a task, its corunners (ITI) has to be decided first



Our approach

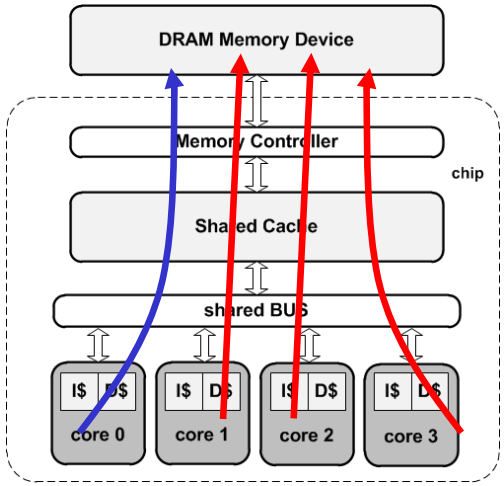
- Carry out execution timing analysis by integrating it with reasoning on task scheduling, in an iterative fashion.

- Focus
  - Partitioned systems comprising a set of *m* independent tasks: T = {t_i,…, t_m}.
  - In each core (scheduling partition) a local dynamic (on-line) scheduler such as EDF, or static (off-line) scheduler can be used.
    - Our proposed approach is orthogonal to the particular local scheduler.
    - We just assume that the local scheduler has an associated schedulability test

- Two approaches
  - Based on fully Time Composable ETB (GR712RC)
  - Based on partially Time Composable ETB (NGMP)

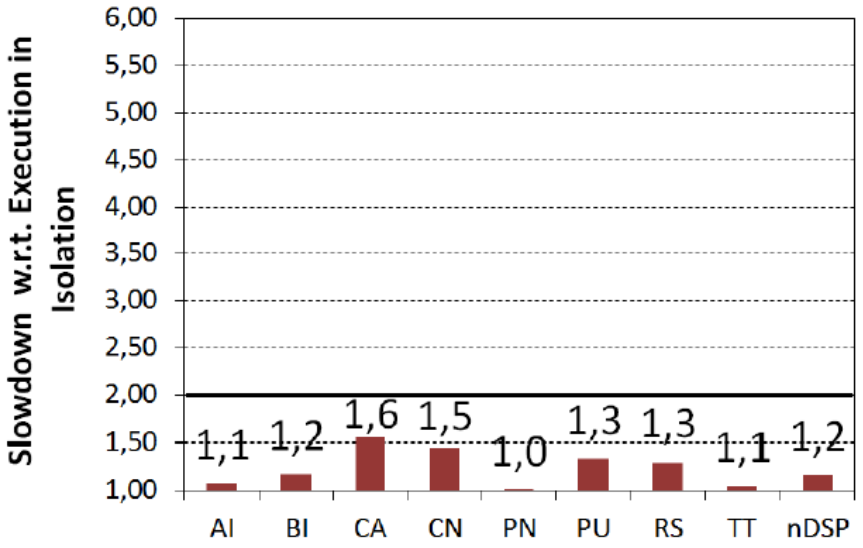
Fully Time Composable ETB

- fTC ETB (C_i^{fTC}): highest execution time observed against RSKs
- Running a given task t_i taking as corunners RSKs
  - Represents a very pessimistic scenario of the inter-task interferences that t_i can experience.
  - No observed real application or benchmark for which another real application introduce more load on observed shared hardware resources than the one introduced by RSK.
- C_i^{fTC} breaks ETA-scheduler dependence
  - Enables time composability
  - May lead to overly pessimistic ETB

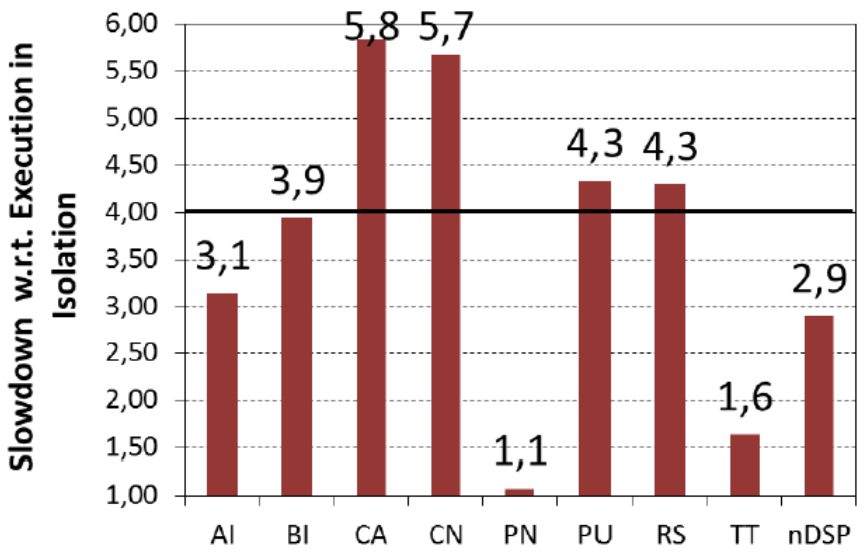


App RSK_1 RSK_2 RSK_3

fTC ETB for allocation?



(b) GR712RC (LEON3)

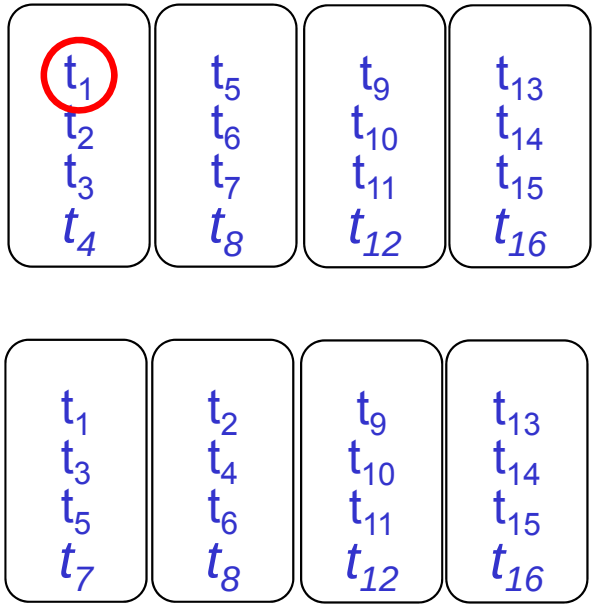


(a) ML510 (LEON4)

- GR712RC → fTC ETB are close to ETB in isolation
  - Use fTC ETB for allocation/scheduling (CPU capacity loss of 23%)
- NGMP → fTC ETB are much higher than ETB in isolation
  - Use pTC ETB for allocation

Partially Time Composable ETB

- Group-mate, Sibling and corunner
- Partially TC (pTC) ETB
  - Each task ti has associated several ETB for each possible task allocation to core, such that the task ETB_i is time composable for a particular allocation ai, of the task set.
    - Group Mates → cannot be corunners of ti
      - gm_{t1} → {t2, t3, t4}
    - Siblings can be corunners of ti
      - sb_{t1} → {t5, t6, …, t16}
    - ETB_{t1} valid for any scheduling of corunners
      - {t5,t9,t11} {t6,t9,t16} {t8,t12,t13}

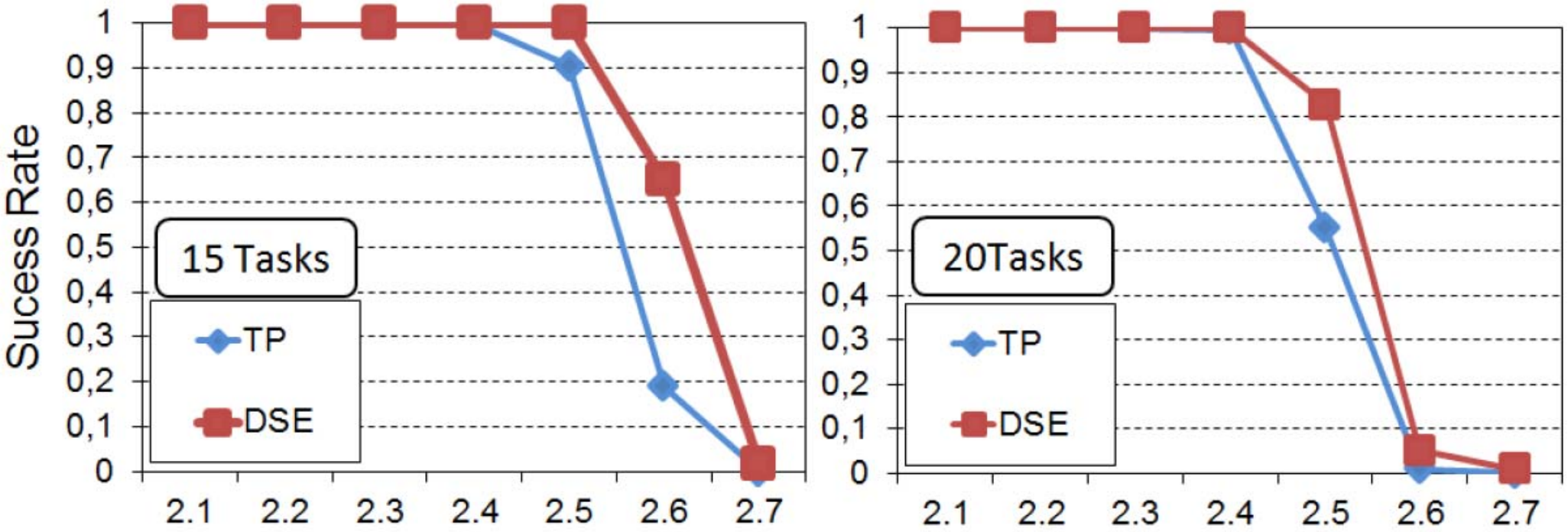


Partially Time Composable ETB

- Before allocating any task
  - For each task we assume its fTC ETB
- Once a task t_i is assigned to a group (core) cj
  - All tasks in cj has a new group-mate and one less potential corunner
    - Adjust the ETB of tasks in cj
    - Adjust the ETB of ti

Timing Analysis with pTC ETB (NGMP)

- 1,000 randomly generated tasksets from EEMBC and nASP
  - 15-task and 20-task workloads
- DSE → Design Space Exploration Approach
- Our pTC-TP provides competitive results

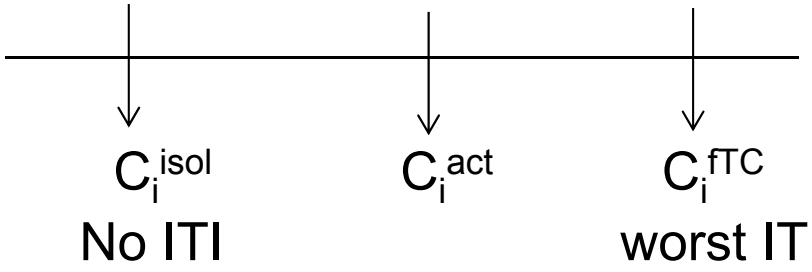


esa

BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

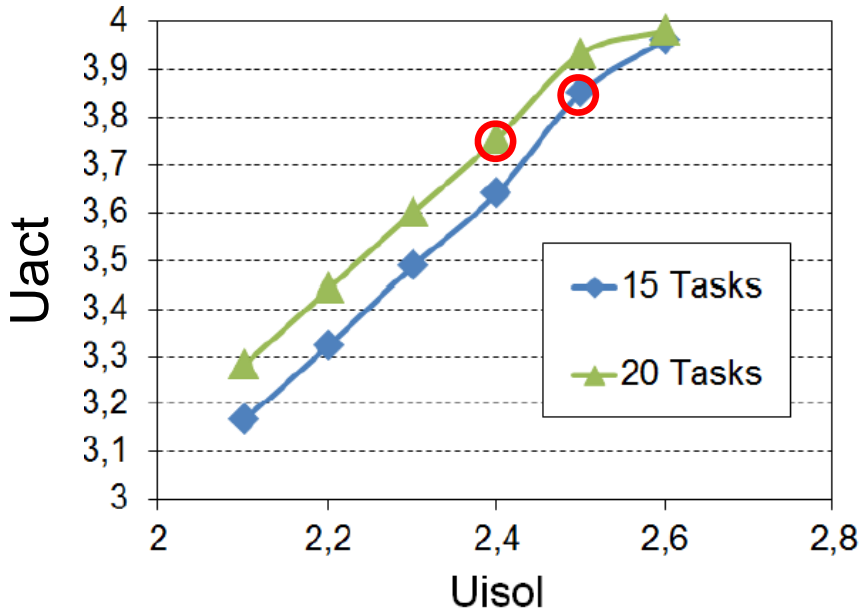
C_i^{isol} vs C_i^{act}

- From previous chart:
  - High success rate for tasksets with U_{wld}^{isol} equal to 2.5 and 2.4 for 15- and 20-task tasksets
- $T=(t_1\ t_2\ t_3\ t_4)$
  - Before allocation
    - $C_i^{isol}/Pi = U_i^{isol} \rightarrow \sum^4_{i=1} U_i^{isol} = U_{wld}^{isol}$
    - $C_i^{fTC}/Pi = U_i^{fTC} \rightarrow \sum^4_{i=1} U_i^{fTC} = U_{wld}^{fTC}$
  - Once allocated
    - $C_i^{act}/Pi = U_i^{act}$
    - $\sum^4_{i=1} U_i^{act} = U_{wld}^{act}$



C_i^{isol} vs C_i^{act}

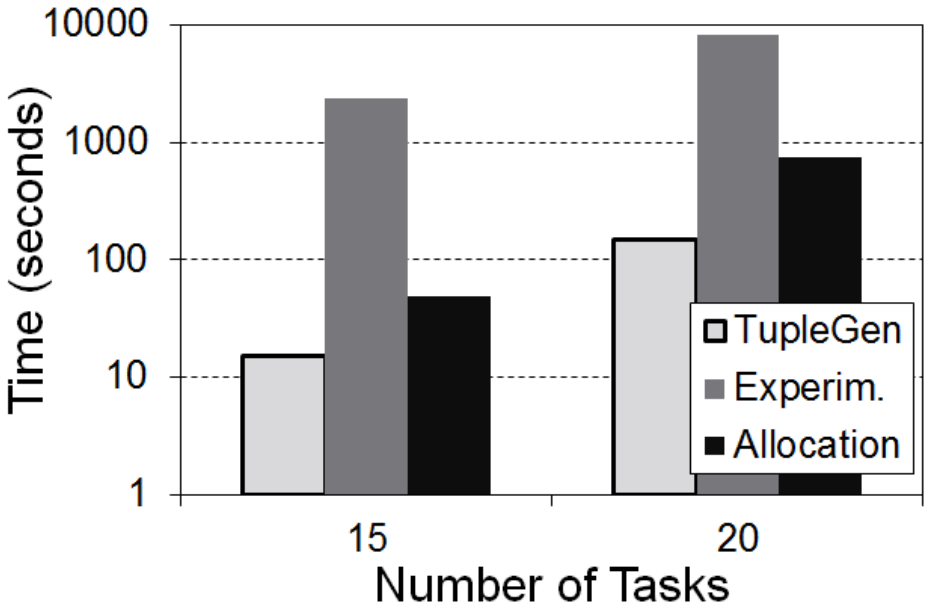
- From previous chart:
  - High success rate for tasksets with U_{wld}^{isol} equal to 2.5 and 2.4 for 15- and 20-task tasksets
  - U_{wld}^{isol} to 2.5 for 15 tasks corresponds to U_{wld}^{act}=3.8
  - U_{wld}^{isol} to 2.4 for 20 tasks corresponds to U_{wld}^{act}=3.7



Maximum Theoretically schedulable capacity (actual utilization) of 4

pTC-allocator: time overheads

- Generation of all tuples comprising all potential combinations (tuples) of {n, n-1, n-2, …, 1} tasks
  - Less than 2 minutes on a Dell Latitude E6420 (Intel Core i7 processor at 2.40GHz)
- The execution of all tuples in our ML510 board takes around 3 hours.
- Finally the execution of pTC-allocator takes less than 17 minutes.



Agenda

- Why multicores in the space domain?
- Inter-task interferences in the NGMP and the GR712RC
- Effect on task scheduling
- **Conclusions and Future work**

Conclusions

- Inter-task interferences have a significant impact on tasks observed execution times on COTS multicores
  - Hard to take them into account in the RTA
- We propose to
  - Integrate timing analysis with task scheduling, in an iterative fashion
  - As a way to handle the circular dependence between them in COTS multicore processors
  - Based on a novel concept of Partially Time Composable (pTC) ETB
- Results
  - GR712RC: we can use fTC ETB
  - NGMP: pTC ETB needed
    - Our approach successfully allocate tasks reducing effect of inter-task interferences

Future work

- Open to carry out similar studies for other boards
  - Contact me
- Implementation of the approach in an RTOS
- Scalability to manycores
- Timing Analysis of COTS multicores with probabilistic methods and tools
  - PROARTIS STREP project
  - PROXIMA IP project
  - ESA-funded activities

Timing Properties of the LEON3-based GR712RC Board. Implications on Task Scheduling

Francisco J. Cazorla, Mikel Fernandez
Jaume Abella, Eduardo Quiñones
Gabriel Fenandez

Marco Zulianello (TO)
Christophe Honvault
Luca Fossati

Francisco J. Cazorla
PhD. Researcher and Director of the CAOS group at BSC
(www.bsc.es/caos) (francisco.cazorla@bsc.es)

Software Systems Division & Data Systems Division
Final Presentation Days
11-12th December 2013, ESTEC