Memory-centric scheduling for phased execution models on multi-core platforms

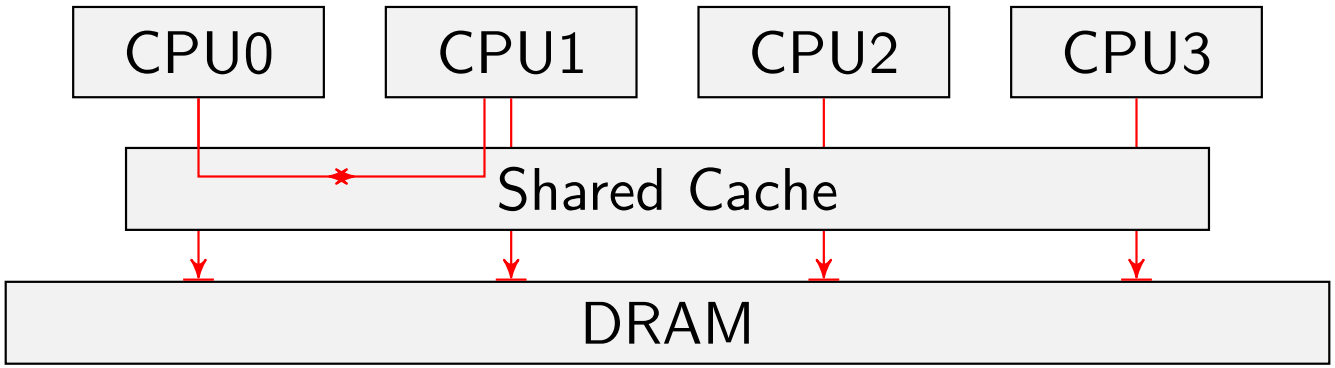
Tomasz Kłoda

LAAS-CNRS / Insa de Toulouse
Toulouse, France

CAPITAL Workshop 2024
14.06.2024

Multiprocessor

Memory interference

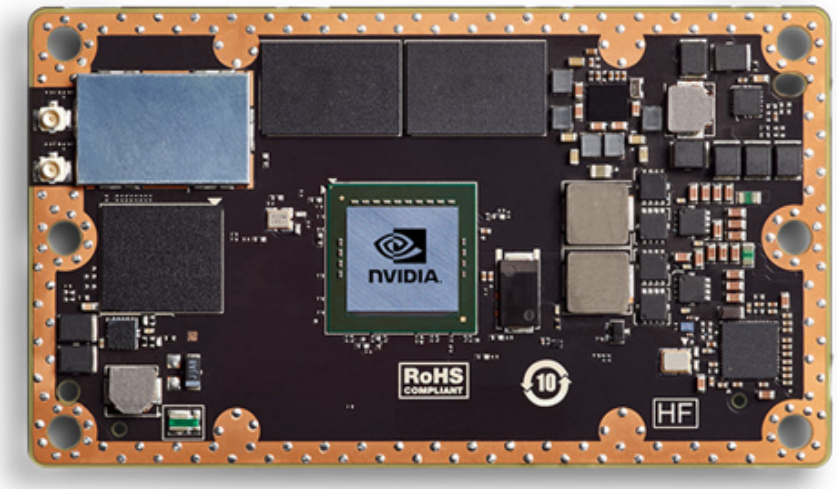


Concurrent accesses to shared hardware resources:

- Shared cache
- DRAM

Commercial-off-the-shelf (COTS) multiprocessors

Nvidia Jetson TX1/TX2



Quad ARM®A57

16-way set associative 2MB L2

Pseudo-random cache

No cache lockdown

No bank partitioning

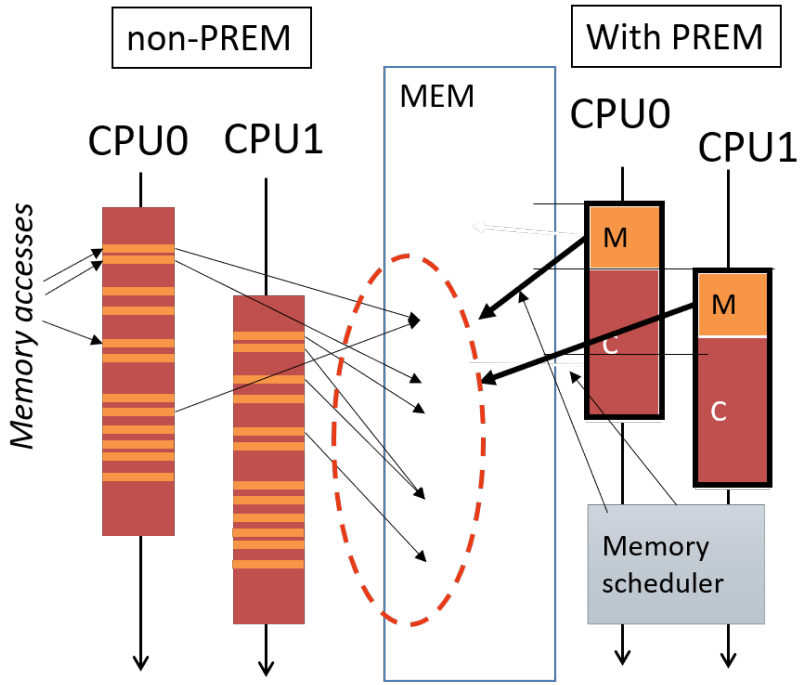
Tomasz Kloda, Marco Solieri, Renato Mancuso, Nicola Capodieci, Paolo Valente and Marko Bertogna.
Deterministic Memory Hierarchy and Virtualization for Modern Multi-Core Embedded Systems. *RTAS 2019*

Paolo Gai, Claudio Scordino, Marko Bertogna, Marco Solieri, Tomasz Kloda and et Luca Miccio
Handling Mixed Criticality on Modern Multi-core Systems: the HERCULES Project. Embedded World 2019

Memory-centric scheduling with multi-phased task models

Acquisition-Execution-Restitution (*AER*) / PRedictable Execution Model (*PREM*)

- Tasks decomposed in several memory-computation phases
- Memory-centric scheduler
- Only one core can access main memory at a time



Guy Durrieu, Madeleine Faugère, Sylvain Girbal, Daniel Gracia Pérez, Claire Pagetti, Wolfgang Puffitsch

Predictable flight management system implementation on a multicore processor (ERTS 2014)

Rodolfo Pellizzoni, Emiliano Betti, Stanley Bak, Gang Yao, John Criswell, Marco Caccamo, Russell Kegley

A Predictable Execution Model for COTS-Based Embedded Systems (RTAS 2011)

Plan

Multi-phase tasks overview

Compiling for multi-phase tasks

Björn Forsberg, Marco Solieri, Marko Bertogna, Luca Benini, Andrea Marongiu

The Predictable Execution Model in Practice: Compiling Real Applications for COTS Hardware *ACM Trans. Embed. Comput. Syst* 2021

Claire Pagetti, Julien Forget, Heiko Falk, Dominic Oehlert, Arno Luppold

Automated generation of time-predictable executables on multi-core. *RTNS* 2018

Frédéric Fort, Julien Forget

Code generation for multi-phase tasks on a multi-core distributed memory platform. *RTCSA* 2019

Renato Mancuso, Roman Dudko, and Marco Caccamo.

Light-prem: Automated software refactoring for predictable execution on COTS embedded systems. *RTCSA* 2014

Zhao Gu, Rodolfo Pellizzoni

Optimizing parallel PREM compilation over nested loop structures. *DAC* 2022

Compiling for multi-phase tasks

Task typologies

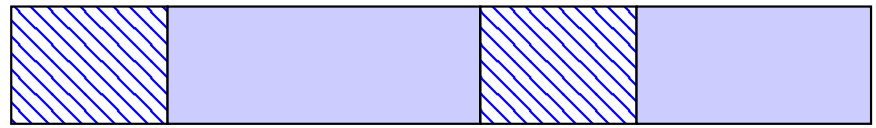
by type of segments:

read-execute

read-execute-write

by number of segments:

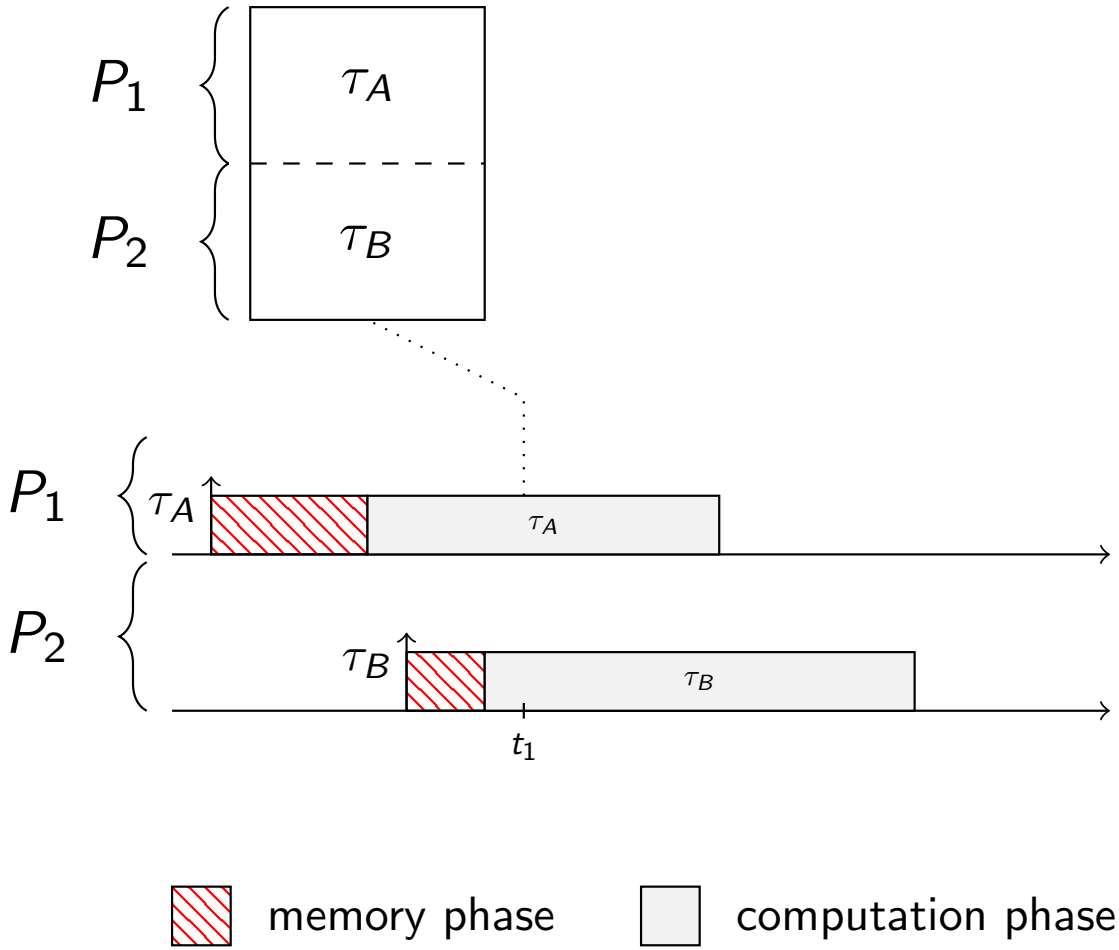
single-segment



multi-segment

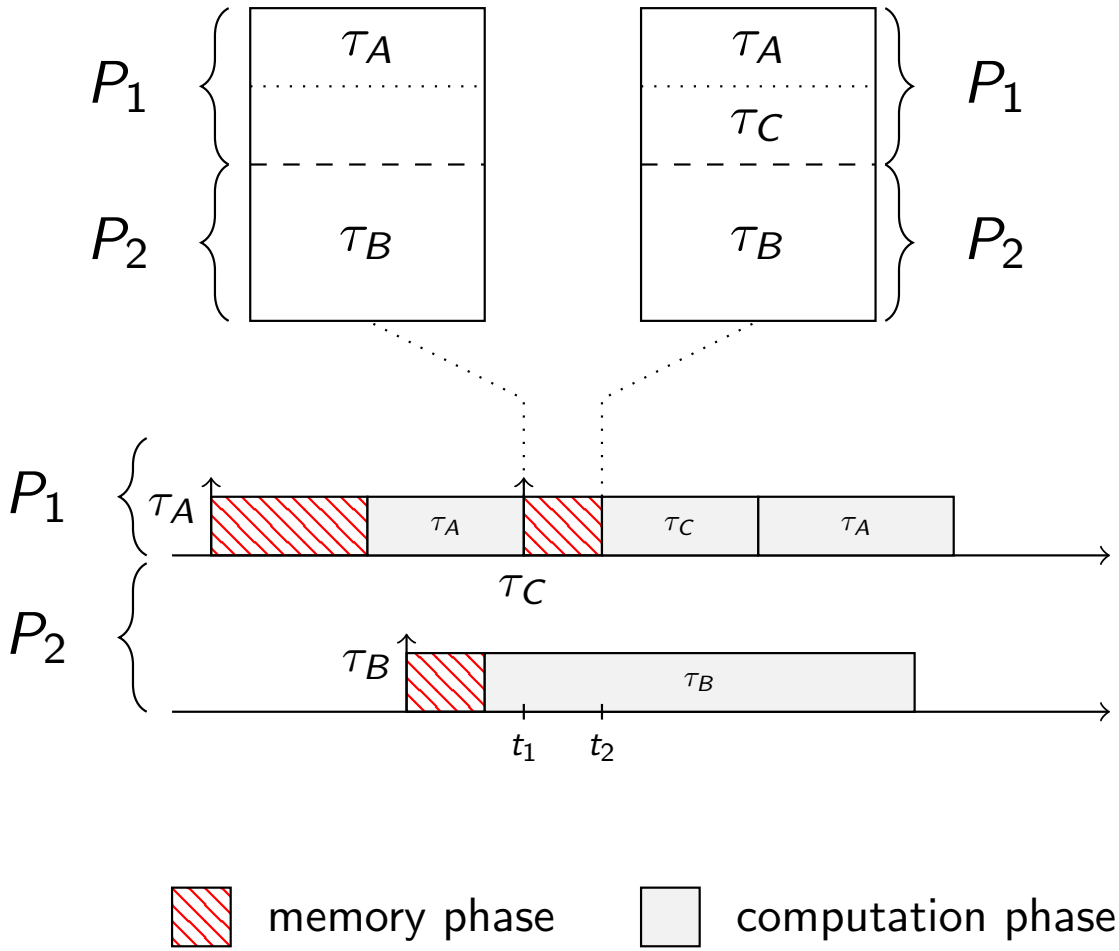
Cache/scratchpad partitioning

Avoid inter-core interference



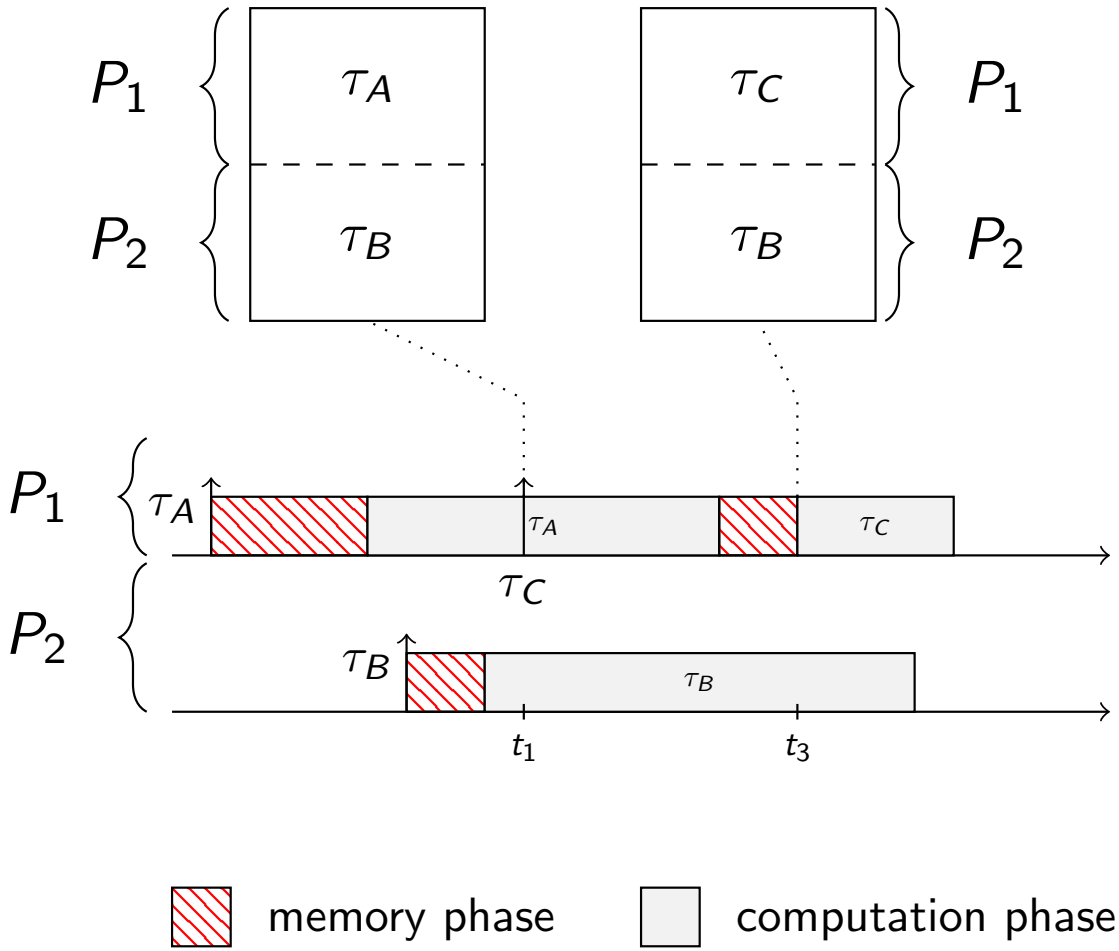
Cache/scratchpad partitioning

Preemptive scheduling



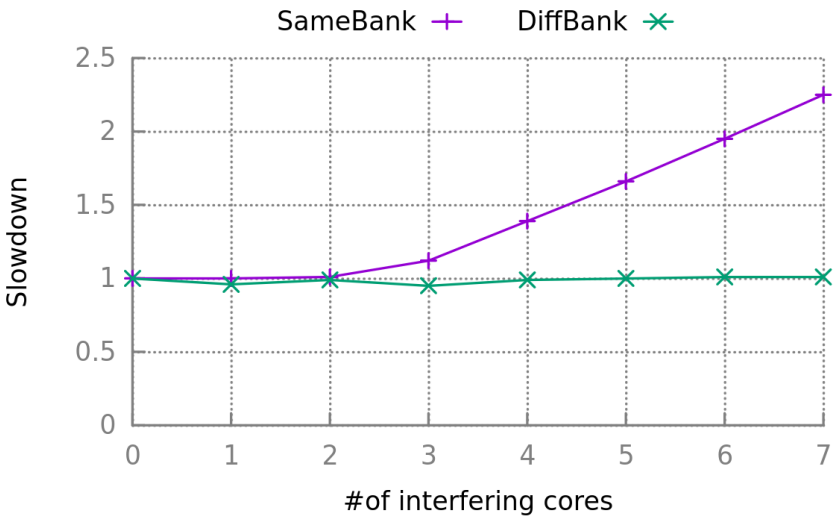
Cache/scratchpad partitioning

Non-preemptive scheduling

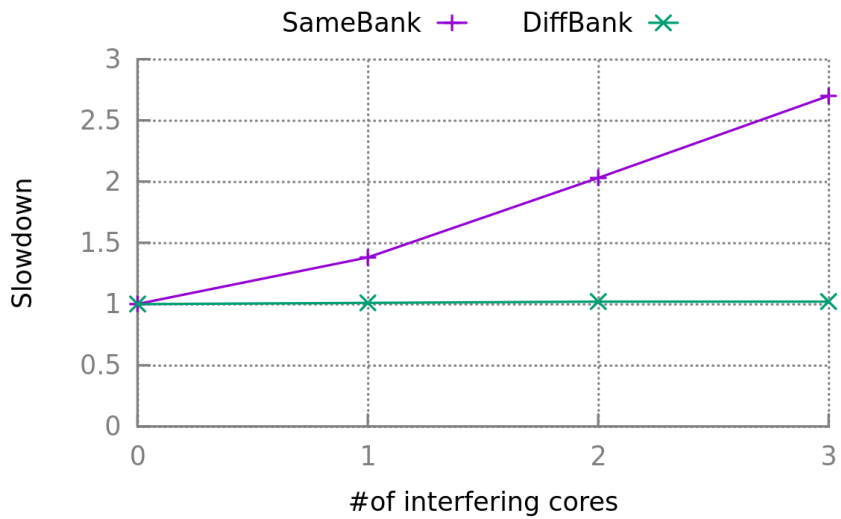


Concurrent accesses

1 out of m vs. k out of m



(a) Freescale P4080



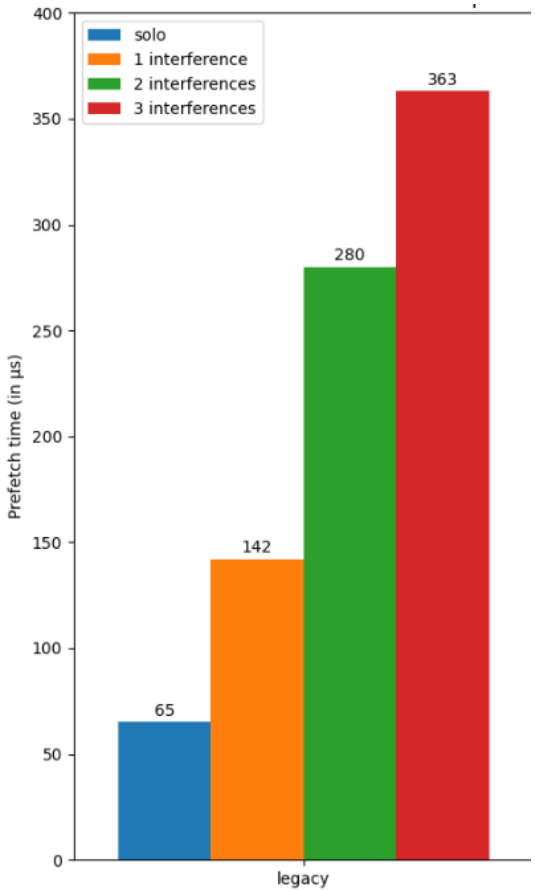
(b) Intel Xeon

Gang Yao, Rodolfo Pellizzoni, Stanley Bak, Heechul Yun, Marco Caccamo

Global Real-Time Memory-Centric Scheduling for Multicore Systems (TC 2016)

Concurrent accesses

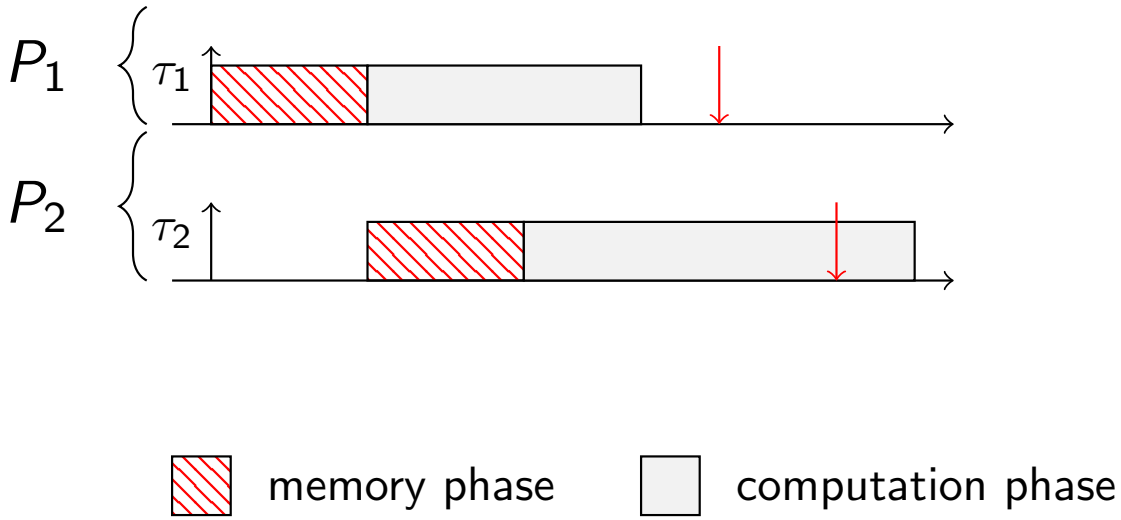
1 out of m vs. k out of m



- Raspberry Pi 4.0 B
- prefetching 200 kB
- with 0, 1, 2 or 3 co-runners

Concurrent accesses

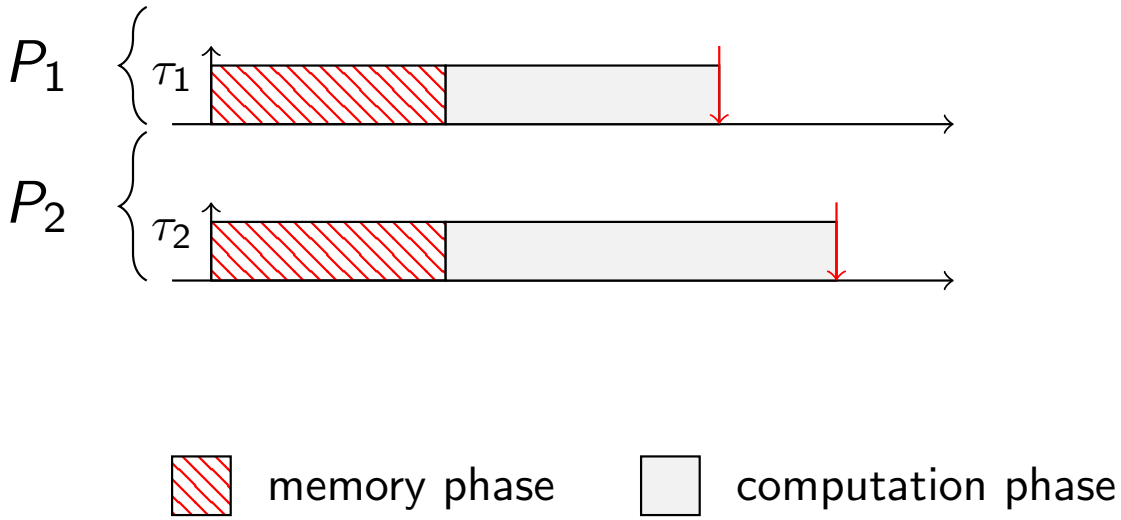
1 out of m vs. k out of m



	mem_i	cmp_i	d_i	$prio_i$
τ_1	4	7	13	high
τ_2	4	10	16	low

Concurrent accesses

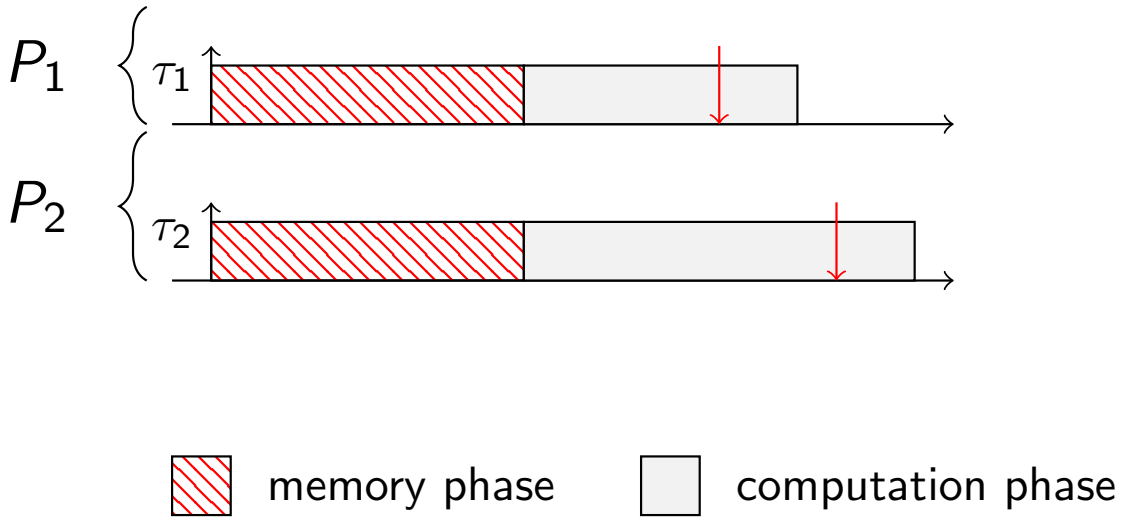
1 out of m vs. k out of m



	mem_i	cmp_i	d_i	$prio_i$
τ_1	**6**	7	13	high
τ_2	**6**	10	16	low

Concurrent accesses

1 out of m vs. k out of m

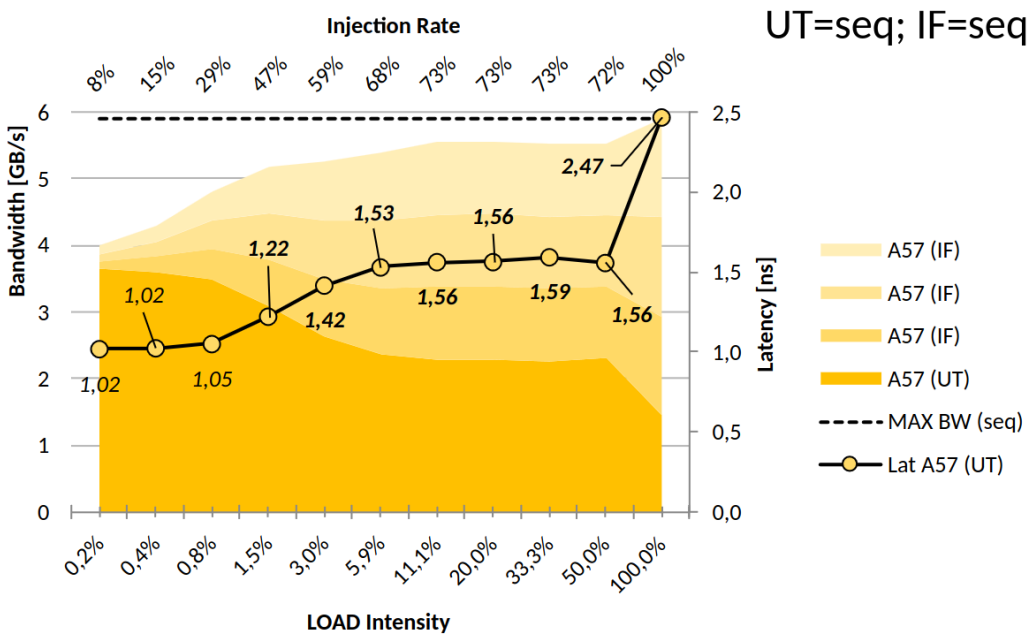


	mem_i	cmp_i	d_i	$prio_i$
τ_1	**8**	7	13	high
τ_2	**8**	10	16	low

Parallel memory phases

Traffic shaping

- higher bandwidth for non-burst workloads
- add mini delays (*e.g.*, `NOPs`) between memory accesses



Roberto Cavicchioli, Nicola Capodieci, Marco Solieri, Marko Bertogna, Paolo Valente, Andrea Marongiu
Evaluating Controlled Memory Request Injection to Counter PREM Memory Underutilization *JSSPP 2020*

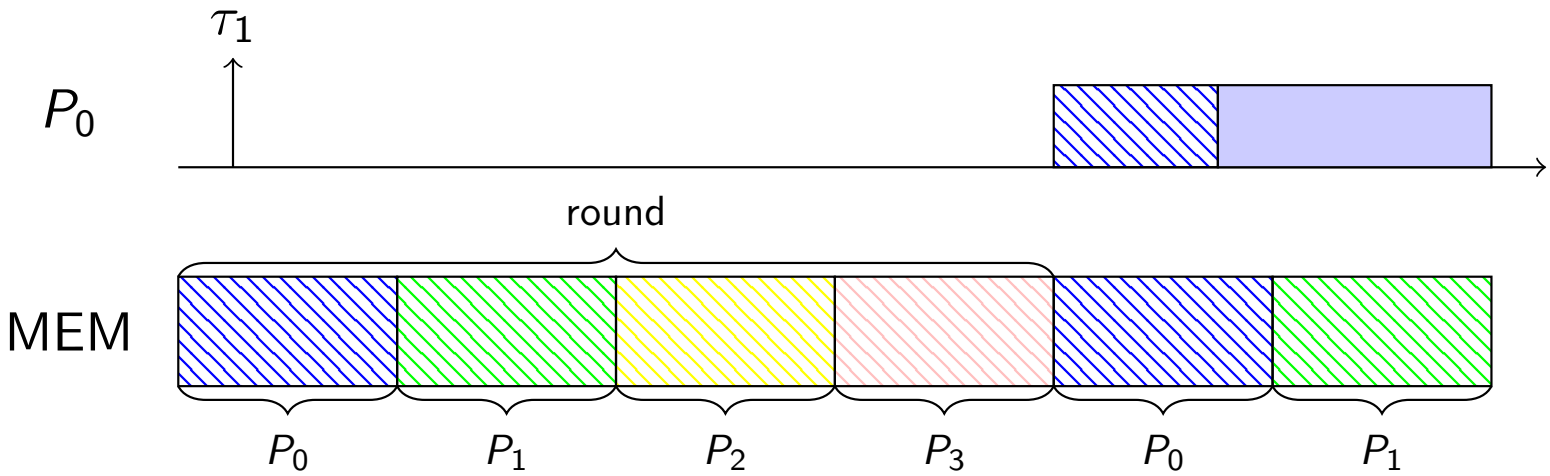
Cédric Courtaud, Julien Sopena, Gilles Muller, Daniel Gracia Pérez
Improving Prediction Accuracy of Memory Interferences for Multicore Platforms RTSS 2019

Dominic Oehlert, Selma Saidi, Heiko Falk
Code-Inherent Traffic Shaping for Hard Real-Time Systems. *ACM Trans. Embed. Comput. Syst. 2019*

Rémi Meunier, Thomas Carle, Thierry Monteil
Correctness and Efficiency Criteria for the Multi-Phase Task Model. *ECRTS 2022*

Memory arbitration

TDMA



Memory arbitration

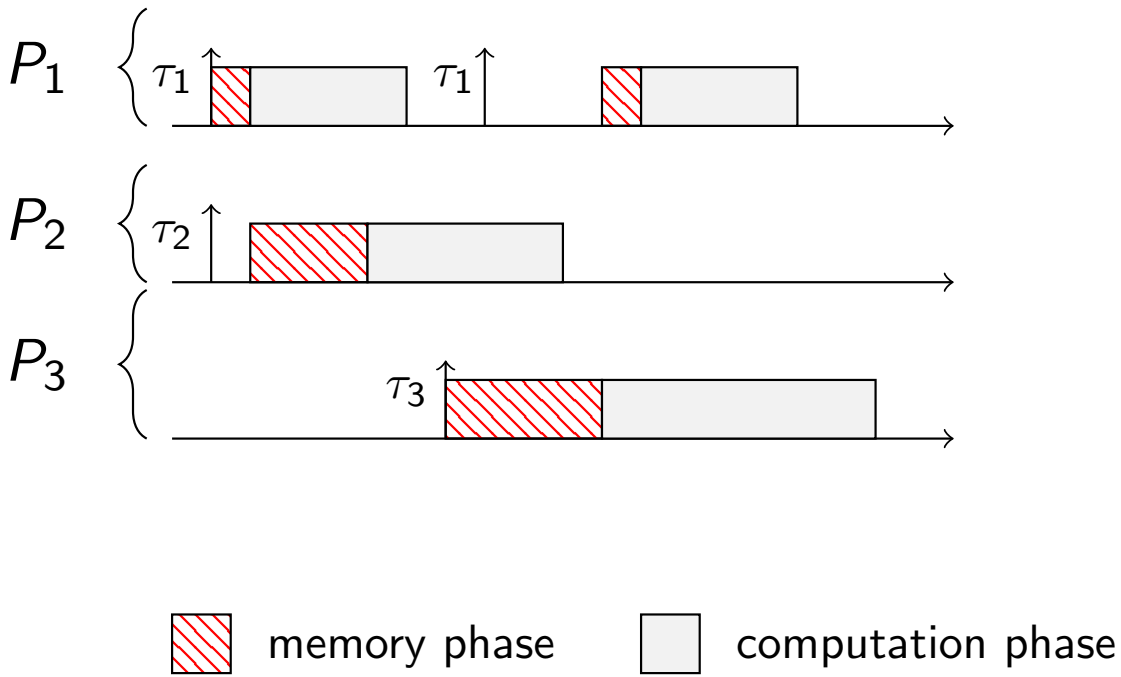
Table driven

- offline
- memory and computation phases fixed at predetermined points
- ILP formulation

Joel Matějka, Björn Forsberg, Michal Sojka, Přemysl Šůcha, Luca Benini, Andrea Marongiu, Zdeněk Hanzálek
Combining PREM compilation and static scheduling for high-performance and predictable MPSoC execution.
Parallel Computing (2019)

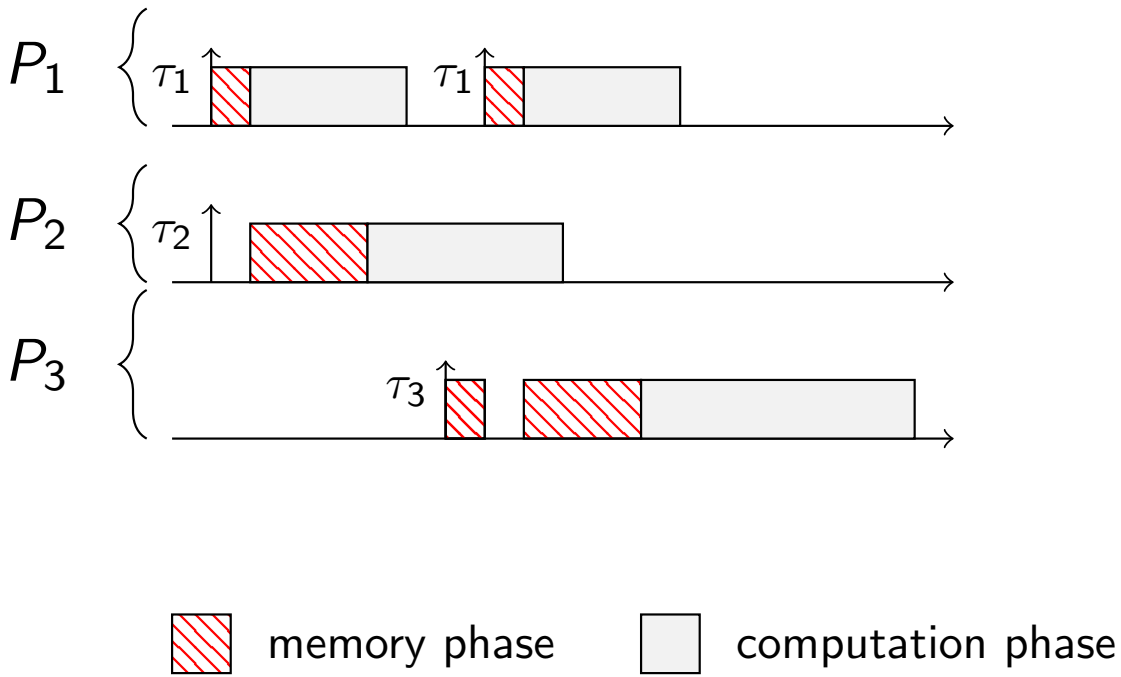
Memory arbitration

Fixed-priority (non-preemptive)



Memory arbitration

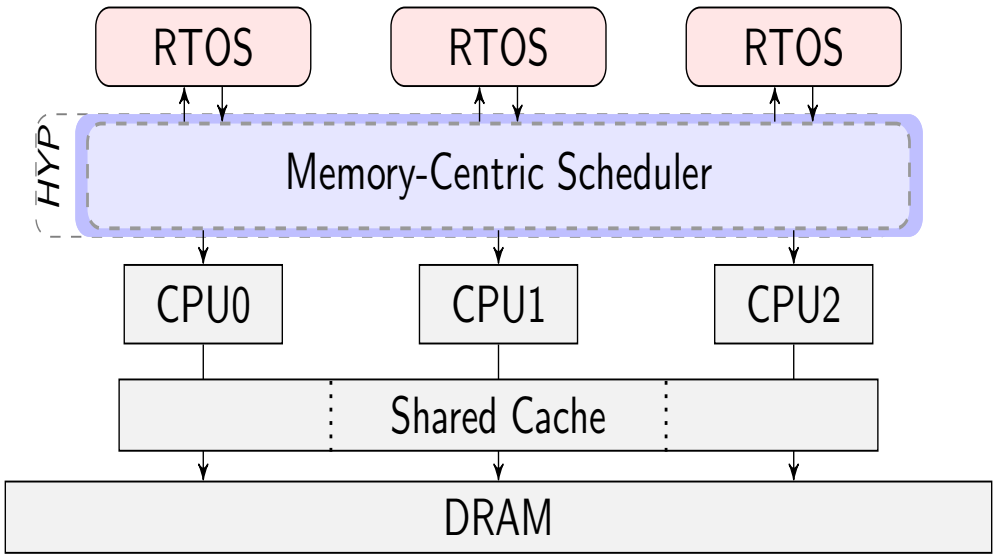
Fixed-priority (preemptive)



Memory-centric scheduler without DMA

Memory-centric scheduler in hypervisor

Preemptive memory phases



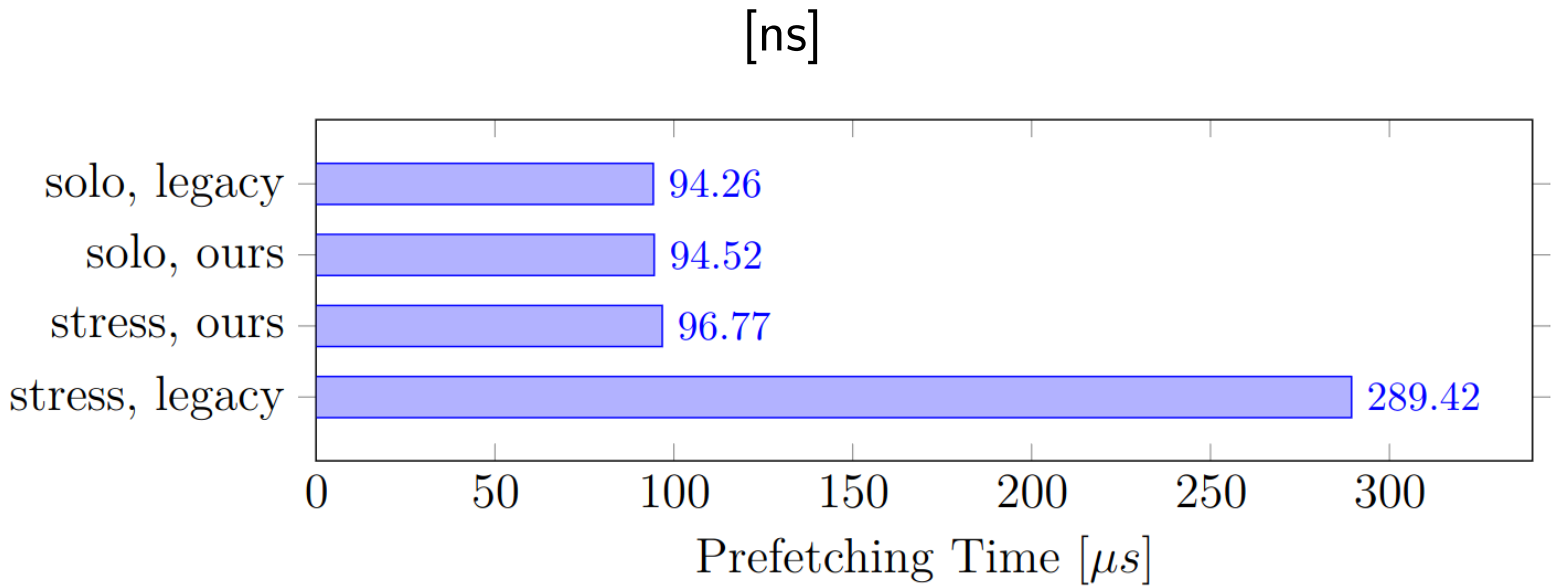
- Jailhouse Linux hypervisor
- single cache partition for each core
- computation phases scheduling: fixed-priority non-preemptive
- memory phases scheduling: preemptive between the cores

Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Memory arbitration

Scheduling overheads

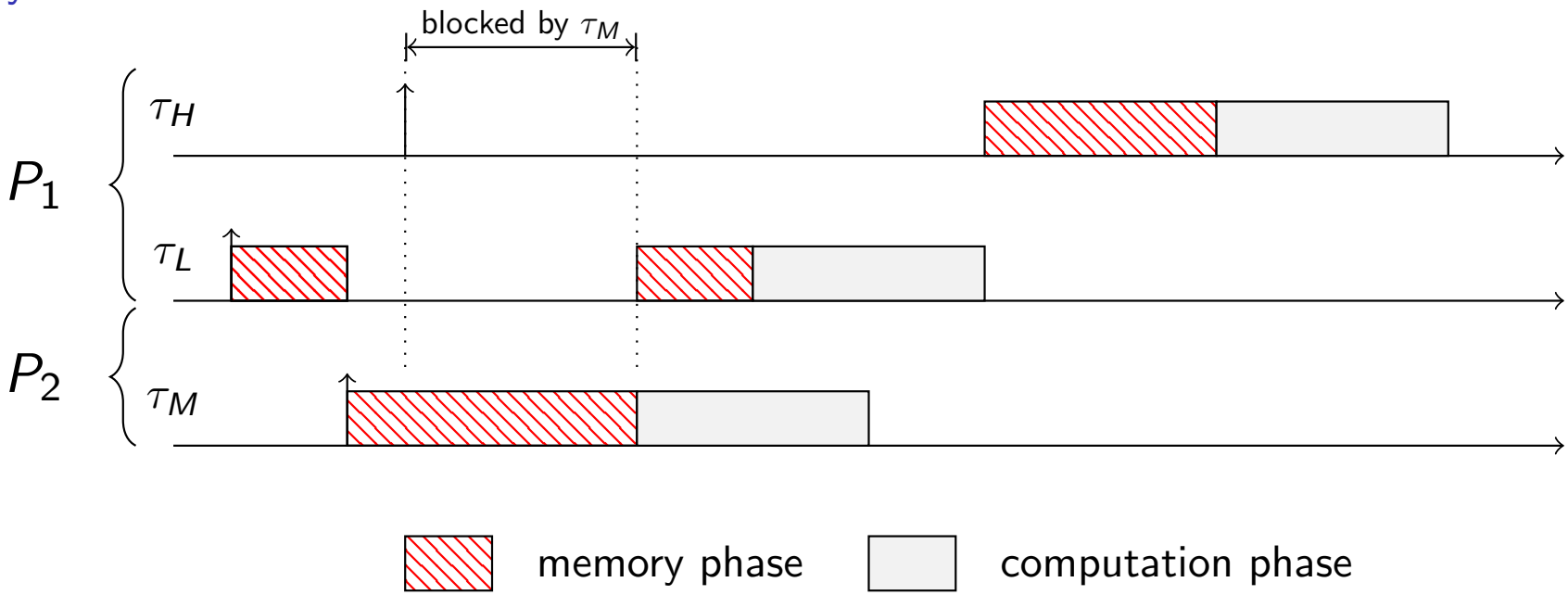
	AVG	STD	WCET	BCET
Hypercall	1265.83	191.50	3129	709
IPI	979.65	80.95	1999	935
Arbitration	127.79	8.94	225	64



Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Priority Assignment

Priority Inversion



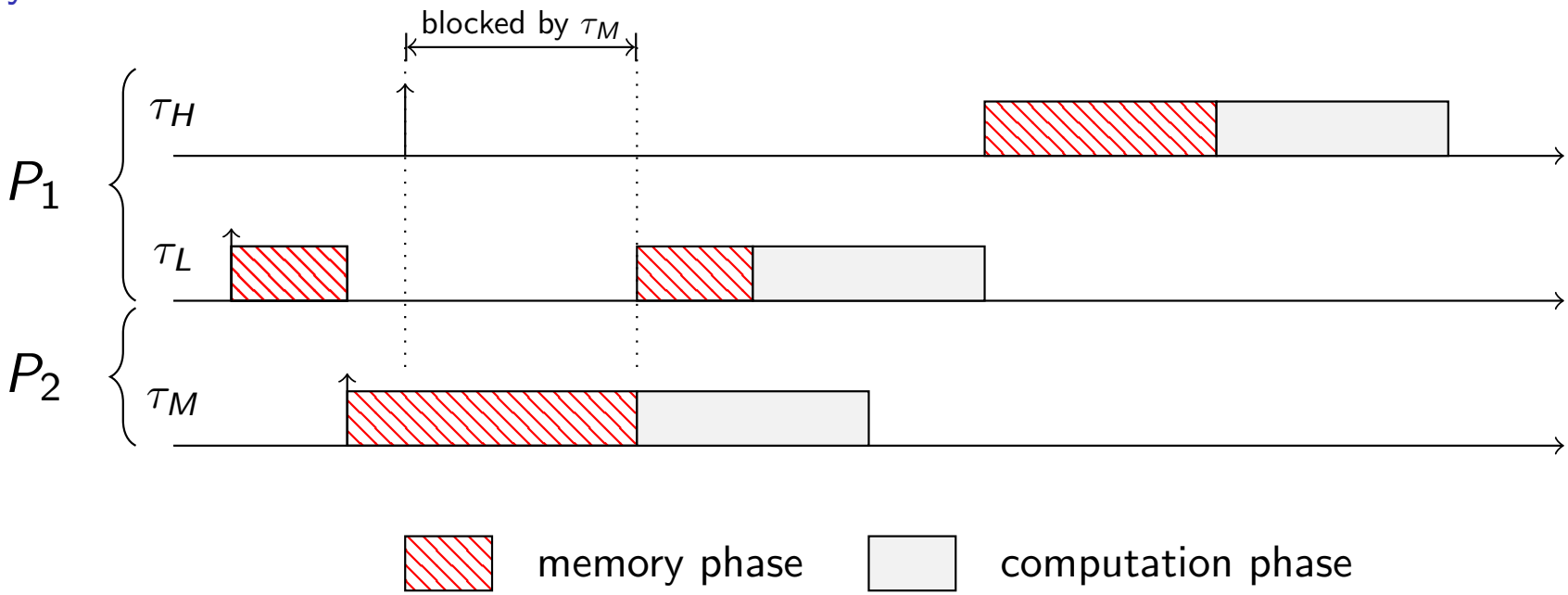
two processors P_1 and P_2

task set: τ_H, τ_L, τ_M with $prio(\tau_H) > prio(\tau_M) > prio(\tau_L)$

processor assignment: $\{\tau_H, \tau_L\} \rightarrow P_1$, $\{\tau_M\} \rightarrow P_2$

Priority Assignment

Priority Inversion



do nothing

priority inheritance

group tasks on processors by priorities (implicit ceiling)

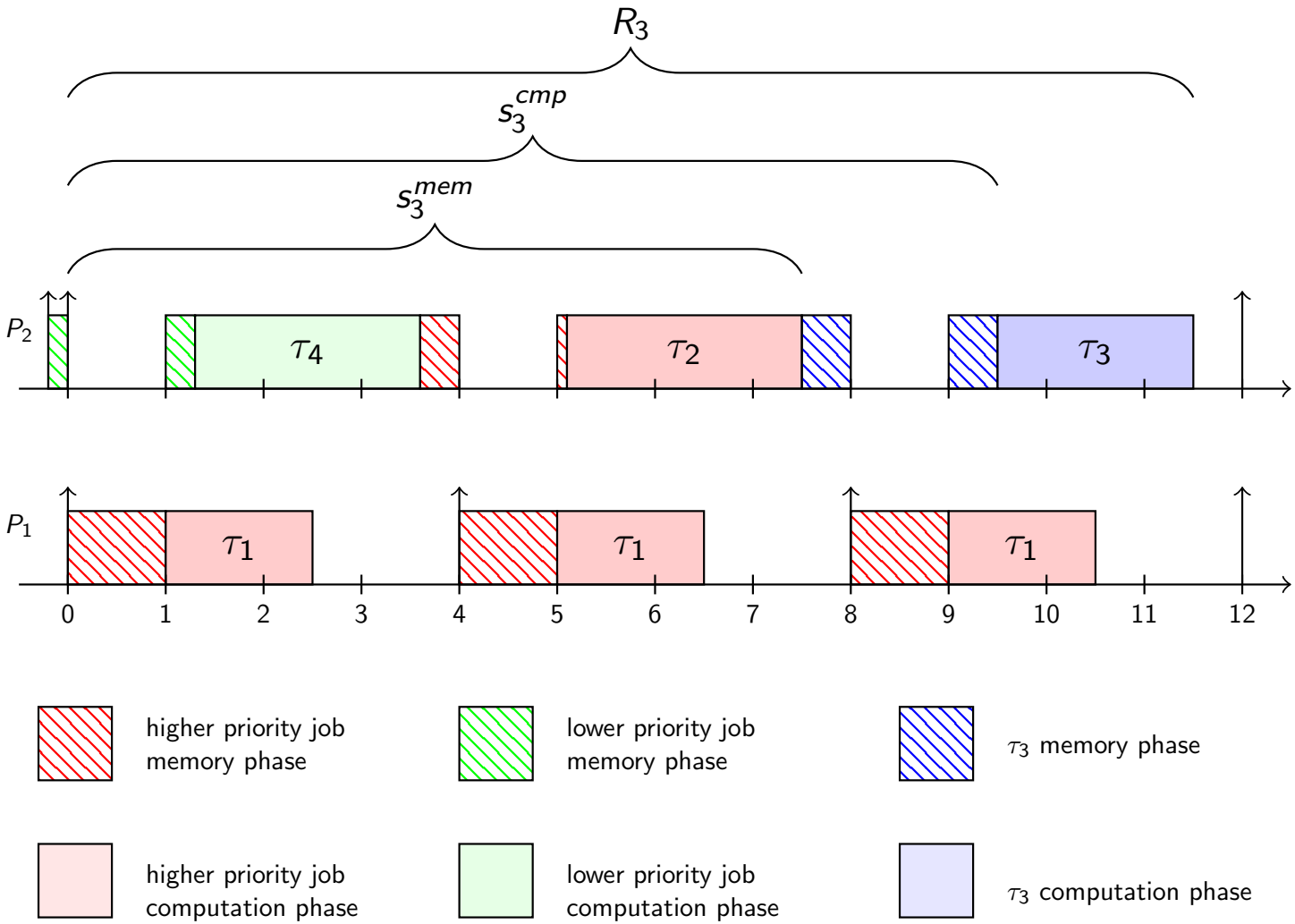
Schedulability analysis

Response Time Analysis example

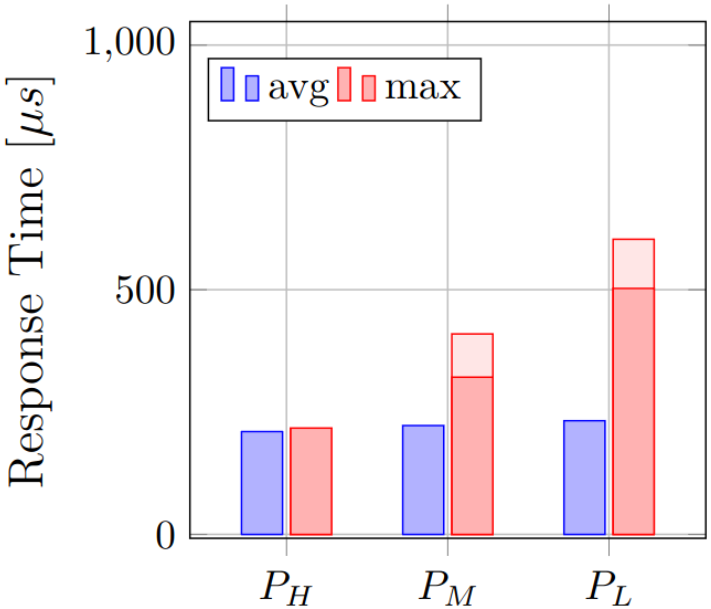
task	priority	period	wcct	wcmt	core
τ_1	highest	4	1.5	1.0	P_1
τ_2	second	12	2.4	0.5	P_2
τ_3	third	12	2.0	1.0	P_2
τ_4	lowest	24	1.7	0.5	P_2

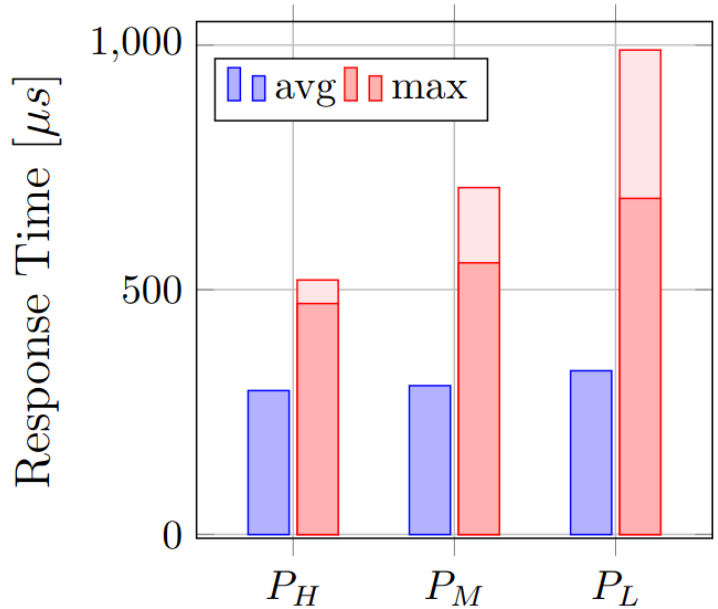
Schedulability analysis

Response Time Analysis example



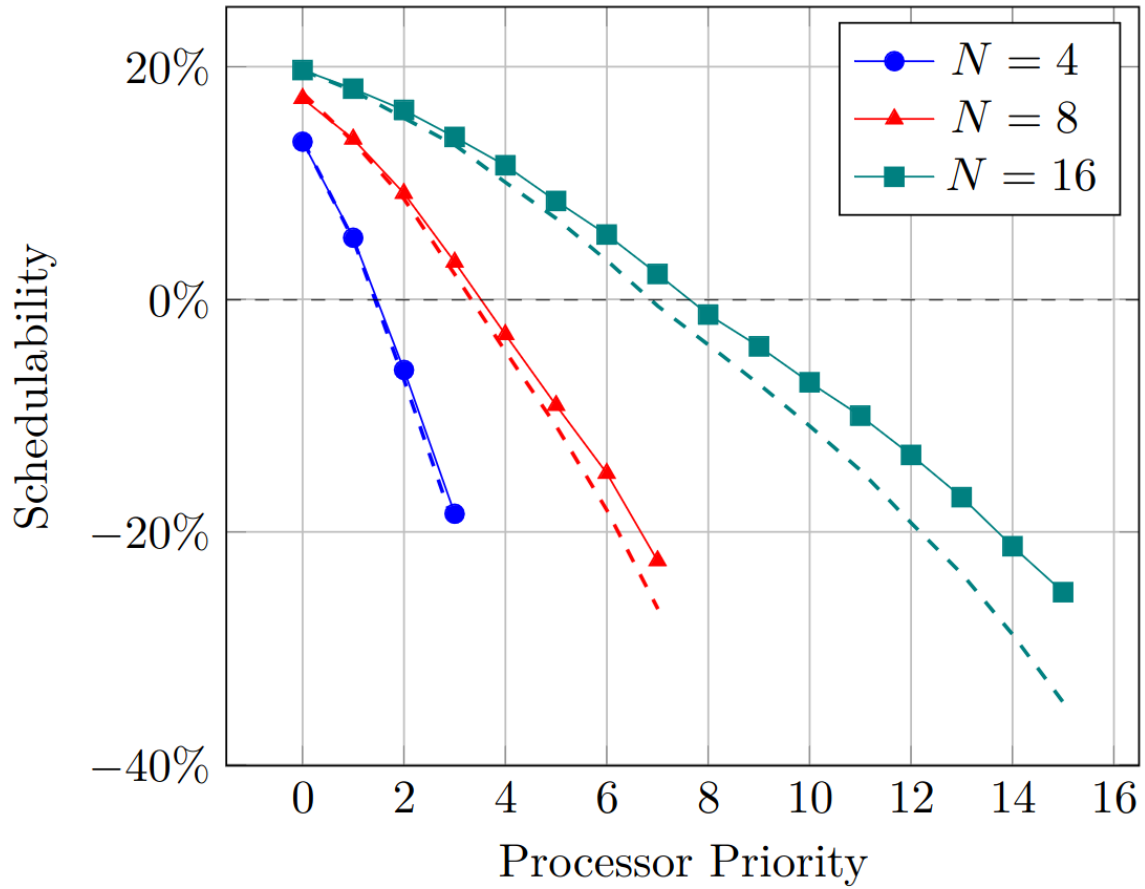
Pessimism in Response Time Analysis





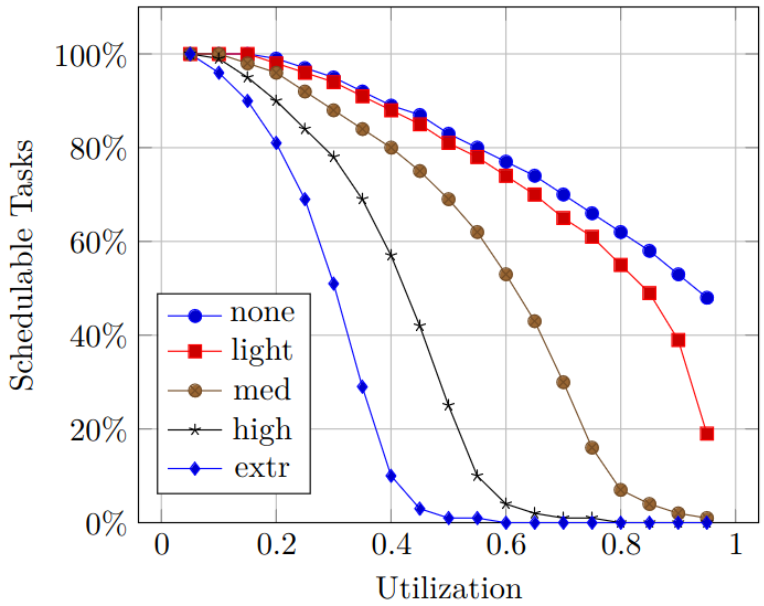
Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Schedulability improvement

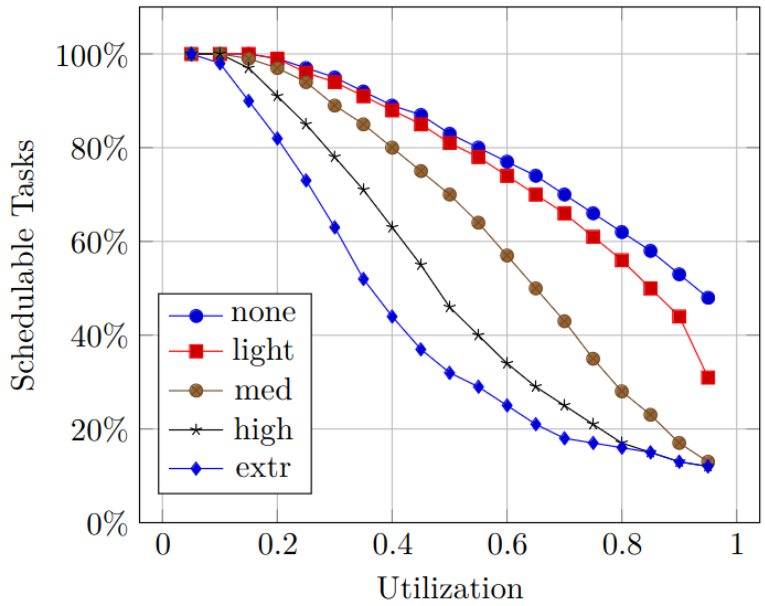


Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Fixed-priority vs. TDMA



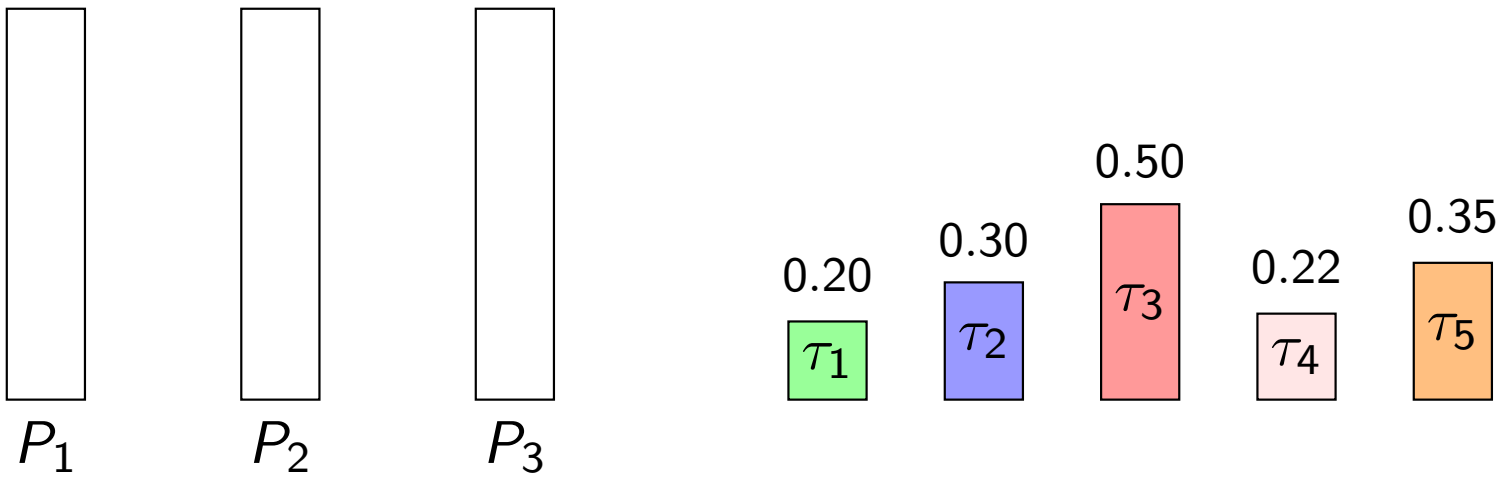
(a) Contention-based scheduling.



(b) Fixed-priority memory-centric scheduling.

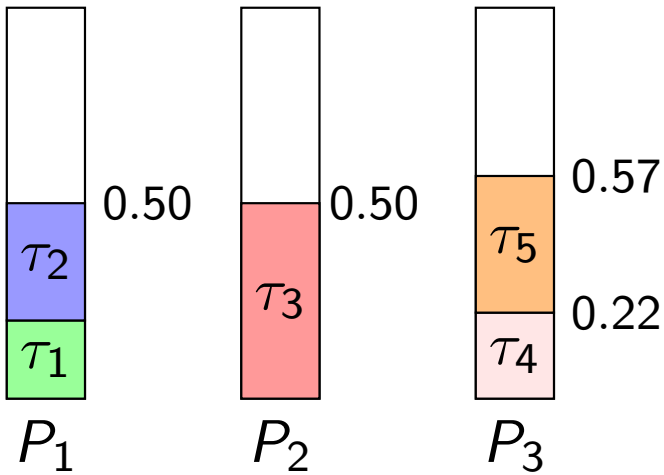
Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Heuristics for task to processor assignment



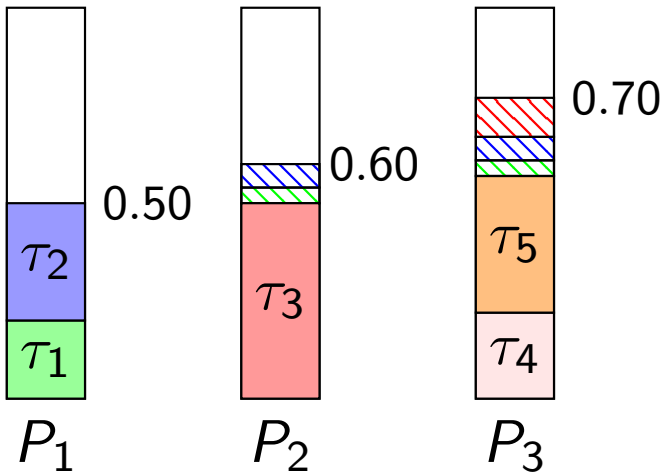
Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Heuristics for task to processor assignment



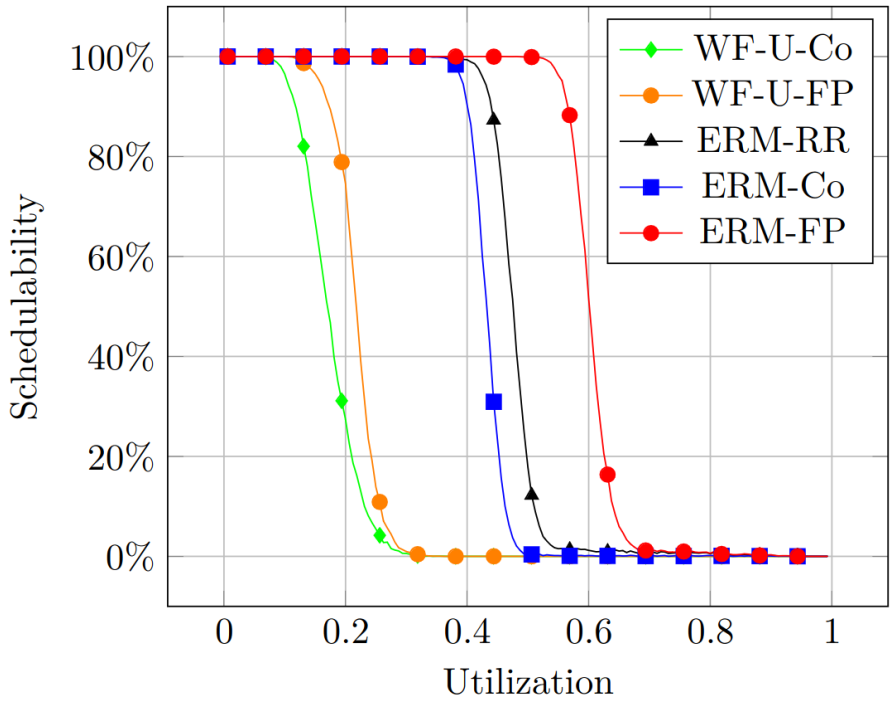
Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Heuristics for task to processor assignment



Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

Heuristics for task to processor assignment



Heuristic *ERM*

- sort tasks in increasing order of periods
- allocate the tasks using First-fit
- cap processor utilization (load balancing)
- tasks' priorities according to *Rate Monotonic*

Gero Schwäricke, Tomasz Kloda, Giovani Gracioli, Marko Bertogna, Marco Caccamo
Fixed-Priority Memory-Centric Scheduler for COTS-Based Multiprocessors. *ECRTS 2020*

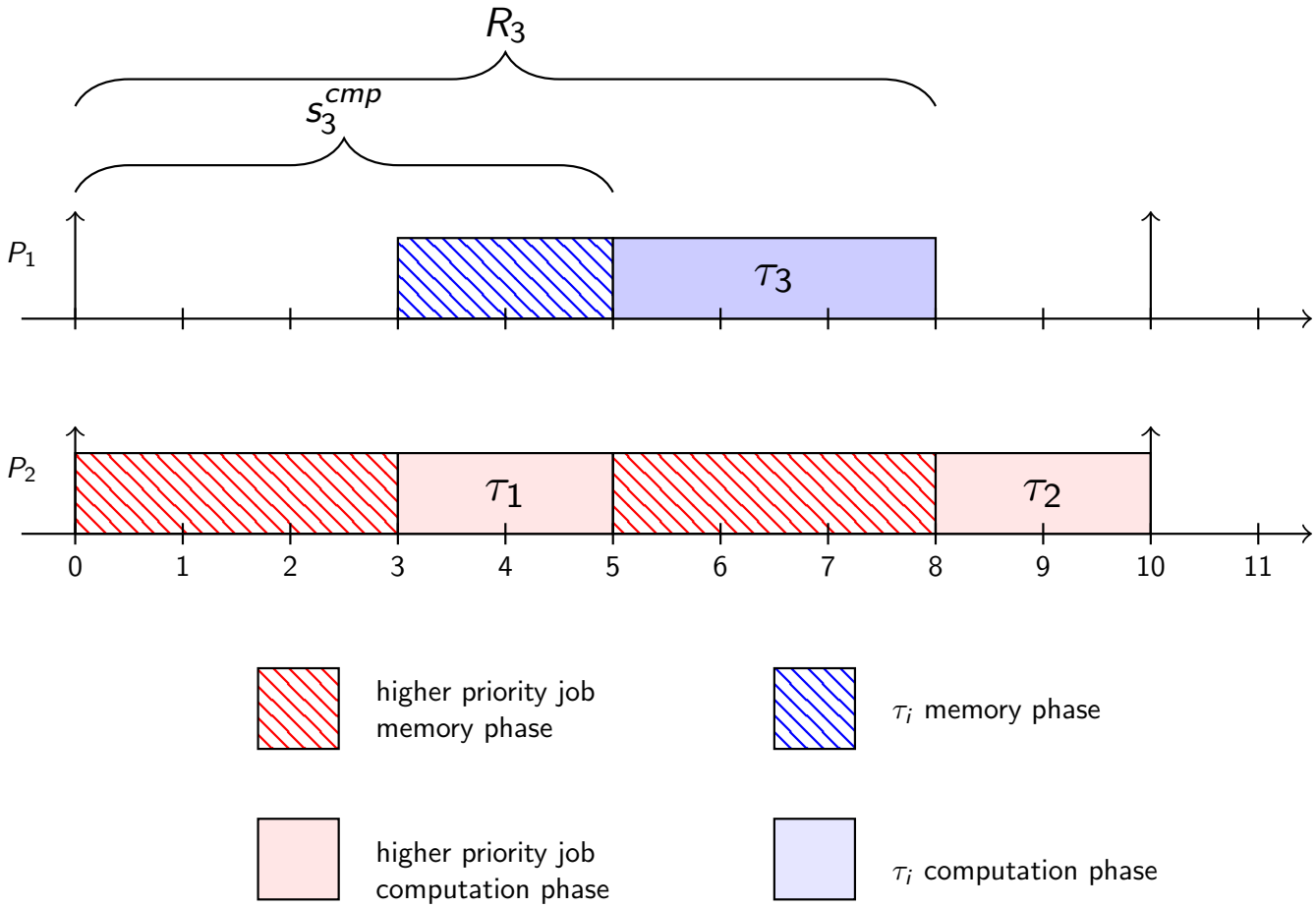
Brice Berna and Isabelle Puaut
PDPA: Period driven task and cache partitioning algorithm for multi-core systems. *RTNS 2012*

Ying Ye, Richard West, Jingyi Zhang, and Zhuoqun Cheng
MARACAS: A Real-Time Multicore VCPU Scheduling Framework. *RTSS 2016*

Drawback of work-conserving memory-centric schedulers

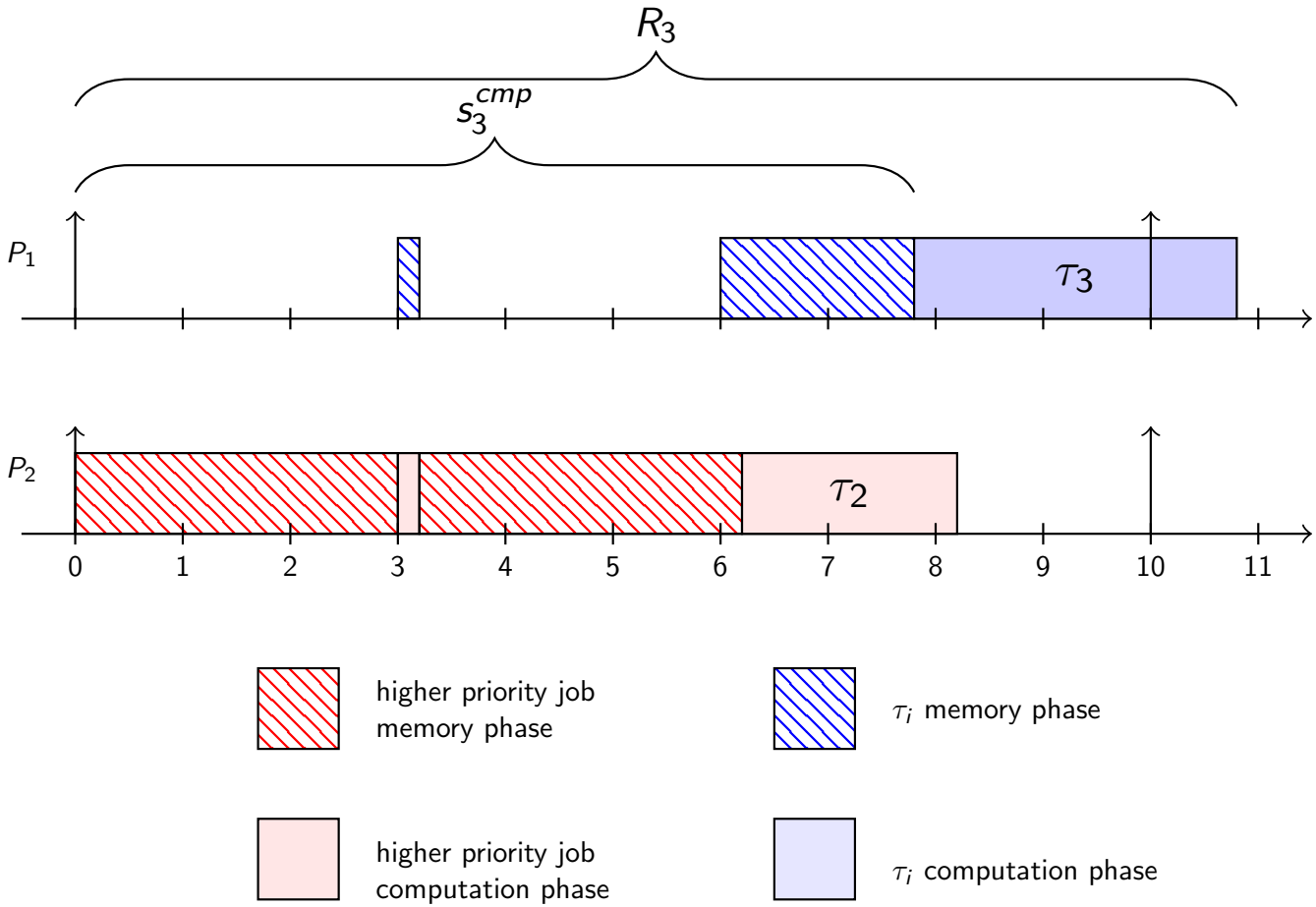
task	priority	period	wcct	wcmt	core
τ_1	high	10	3	2	P_2
τ_2	medium	10	3	2	P_2
τ_3	low	10	2	3	P_1

Drawback of work-conserving memory-centric schedulers



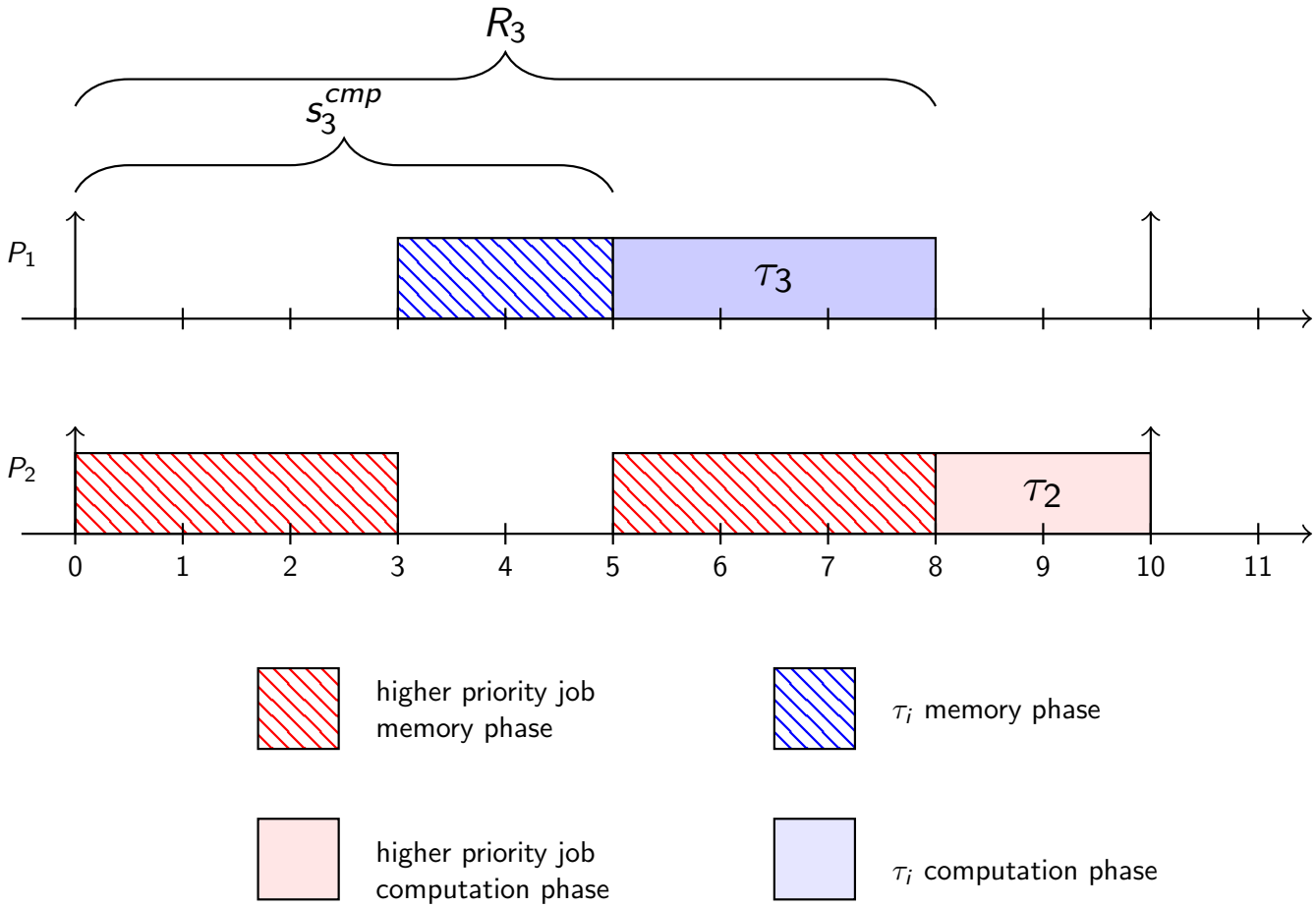
Tasks execute with their worst-case parameters

Drawback of work-conserving memory-centric schedulers



Task τ_1 completes its computation phase immediately

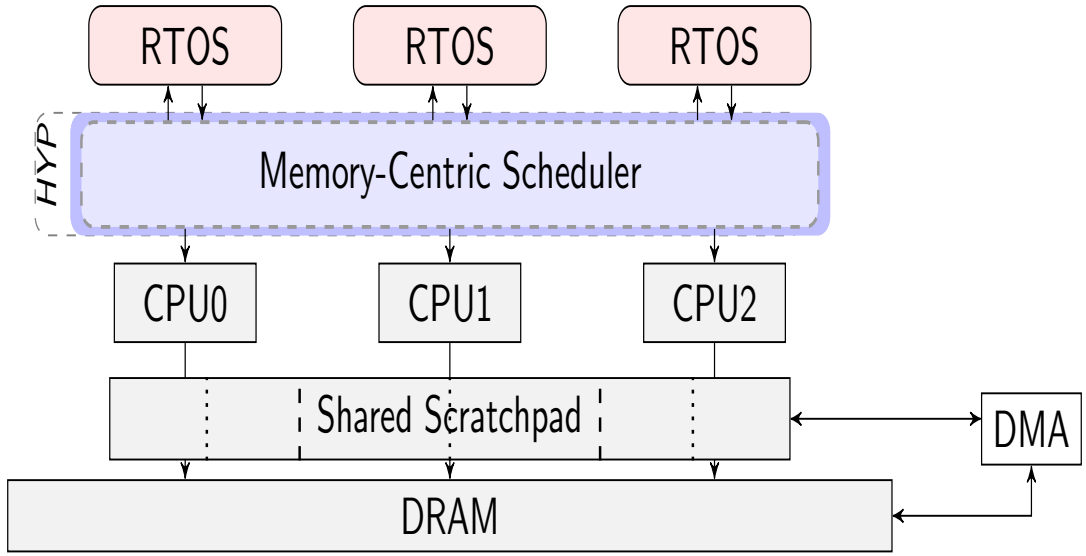
Drawback of work-conserving memory-centric schedulers



Throttle memory requests

Memory-centric scheduling with DMA

Memory-centric scheduling with DMA

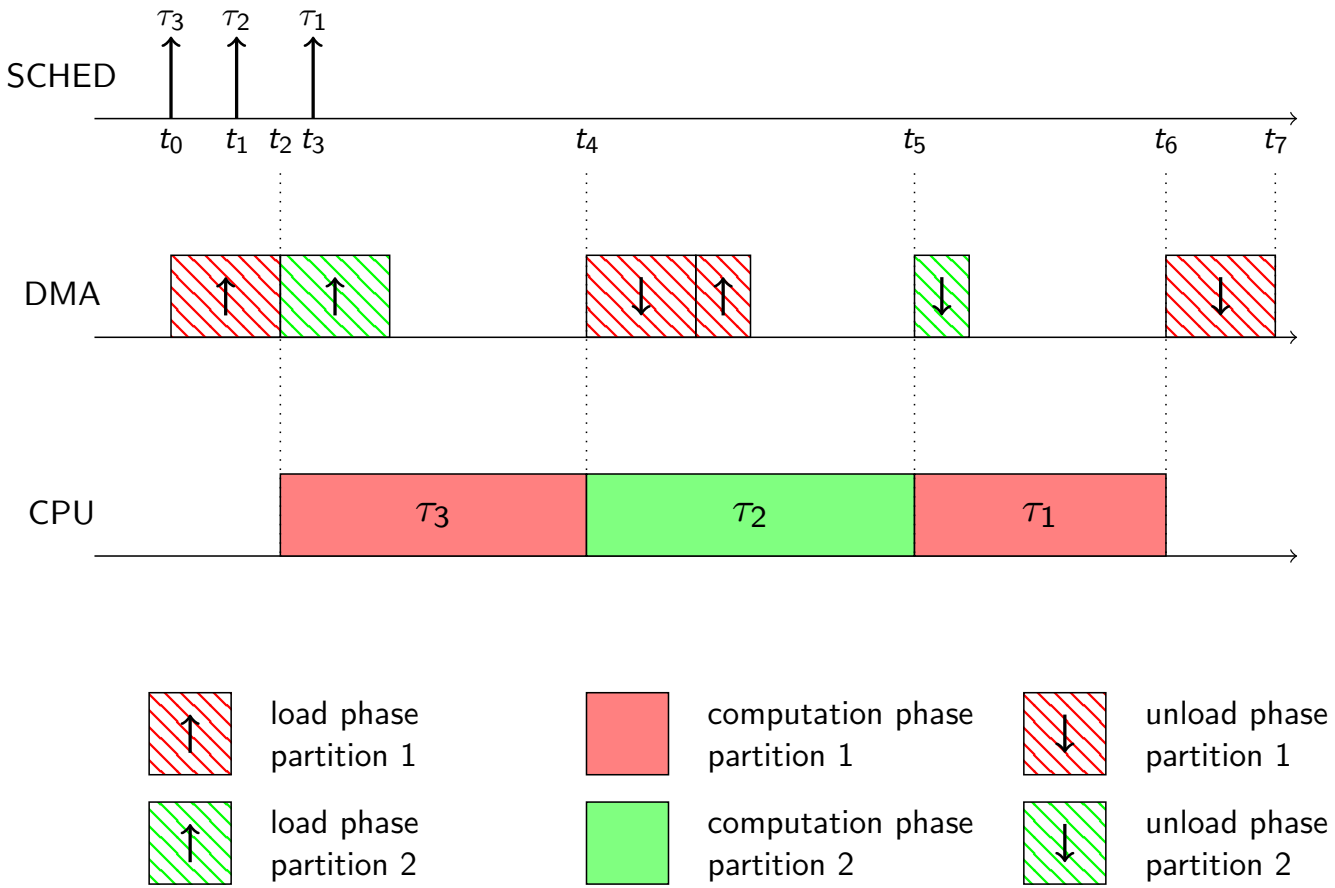


- Two scratchpad partitions for each core
- Non-preemptive scheduling at core-level
- Memory arbitration with *TDMA*
- Read-Execute-Write task model

T. Kloda, G. Gracioli, R. Tabish, R. Mancuso, R. Mirosanlou, R. Pellizzoni, M. Caccamo
Scheduling and System Design for Real-Time Applications on Heterogeneous MPSoC Platforms.
ACM TECS 2023

Memory-centric scheduling with DMA

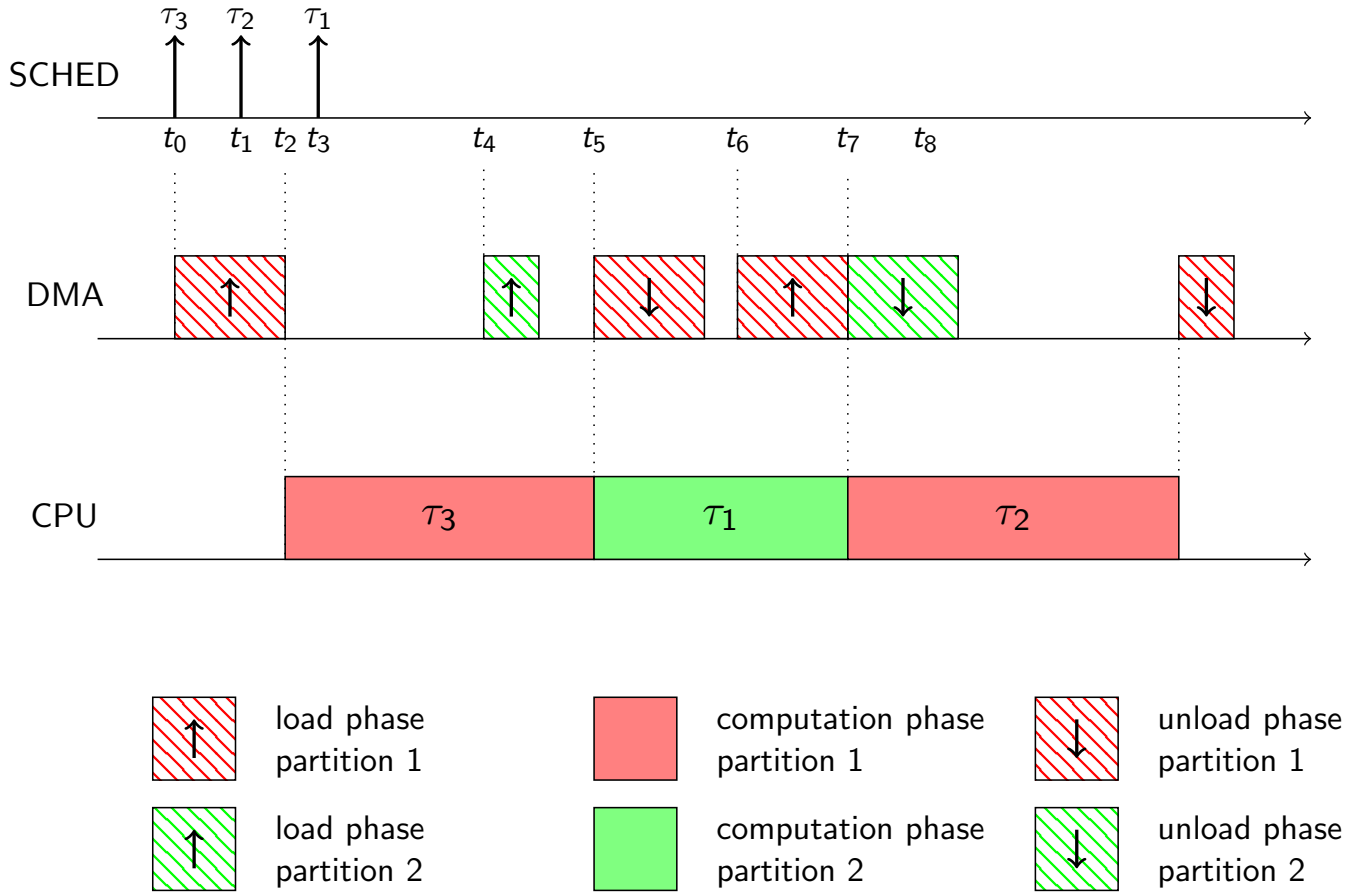
Early Load (state-of-the-art work-conserving scheduler)



R. Tabish, R. Mancuso, S. Wasly, S. S. Phatak, R. Pellizzoni et M. Caccamo,
A Reliable and Predictable Scratchpad-centric OS for Multi-core Embedded Systems. RTAS 2017

Memory-centric scheduling with DMA

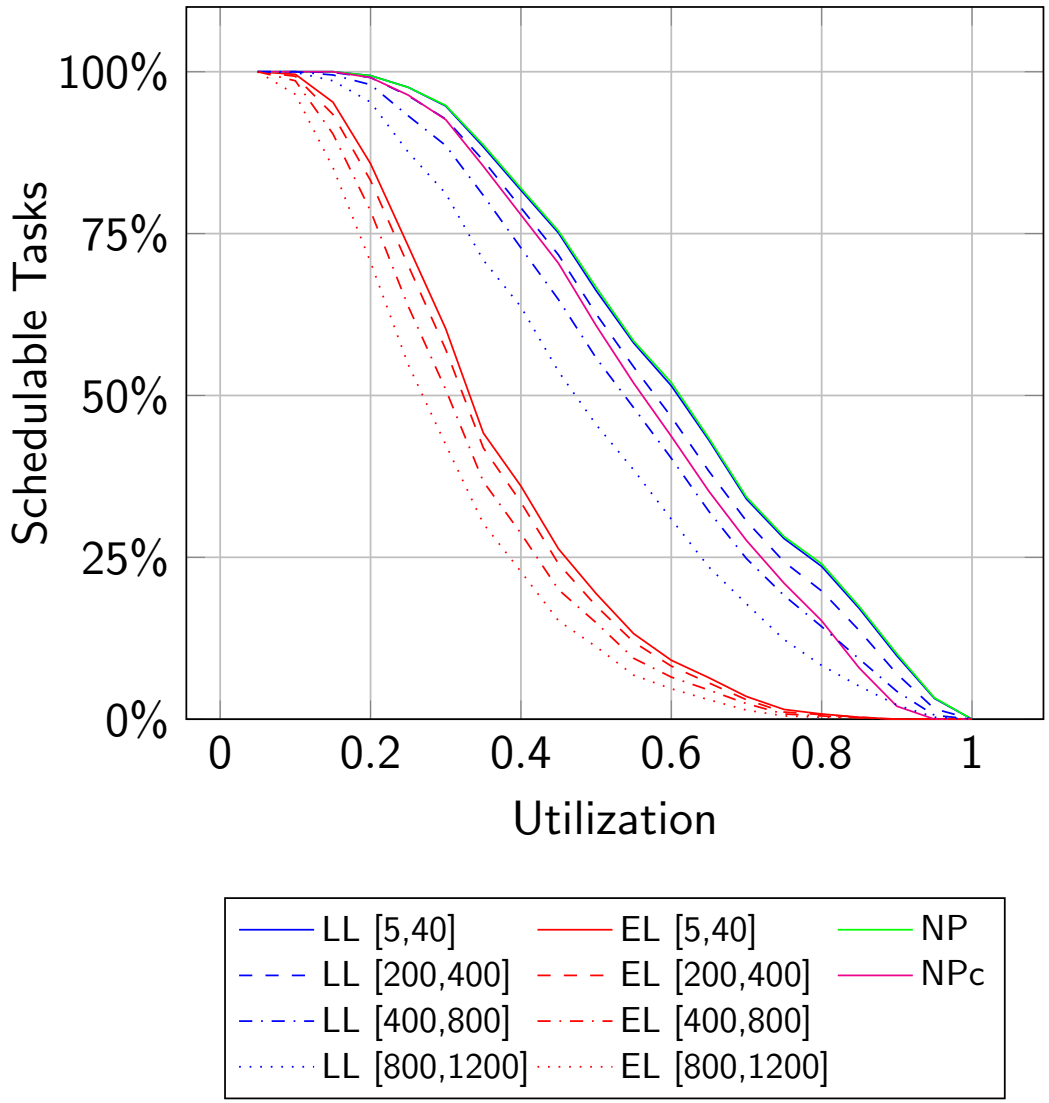
Lazy Load



T. Kloda, G. Gracioli, R. Tabish, R. Mancuso, R. Mirosanlou, R. Pellizzoni et M. Caccamo
Scheduling and System Design for Real-Time Applications on Heterogeneous MPSoC Platforms.
ACM TECS 2023

Memory-centric scheduling with DMA

Experiments



$[s_{min}, s_{max}]$ memory phase duration in μs

LL - Lazy Load, EL - Early Load (state-of-the-art)

Conclusions

- new *models* for better parallelism
- new *compilers* and *sensitivity* analyses
- new *schedulers* beyond work-conserving algorithms
- new *hardware* features (Arm's *MPAM* and Intel's *RDT*)

Merci de votre attention