Run-time adaptation of task execution in time-critical systems: Challenges and Solutions

Angeliki Kritikakou

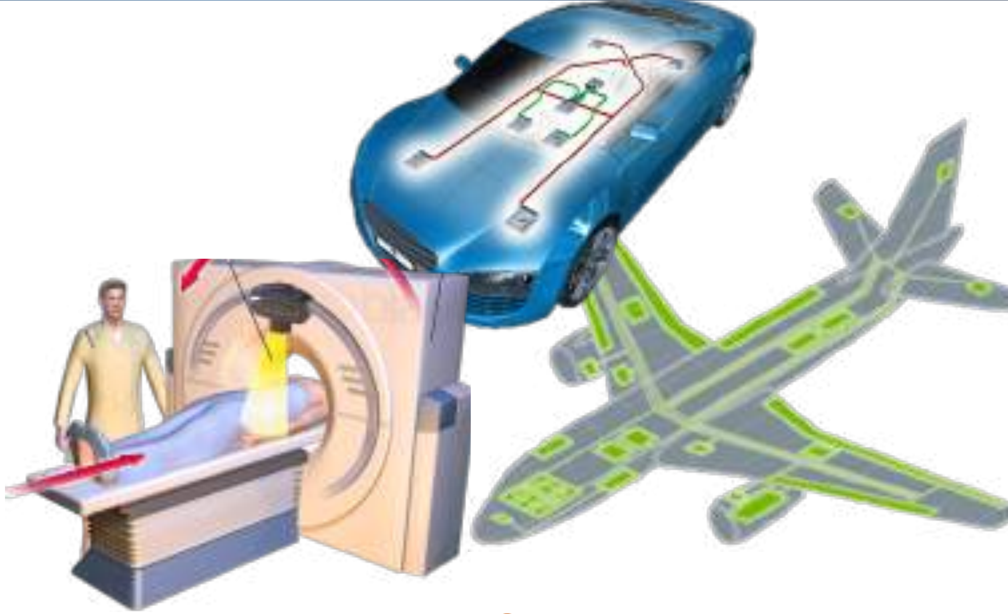
Univ Rennes, IRISA, Inria, CNRS

June 2021

Institut de Recherche en Informatique et Systèmes Aléatoires

Time-critical systems

- **Application domains**
  - Avionics (Fly-by-wire)
  - Automotive (Airbag)
  - Medical (X-Ray)



- **Common characteristics**
  - **C**riticality **L**evel: dual-criticality model
  - **R**eal-**T**ime constraint: Hard or soft (no)
  - **P**riority

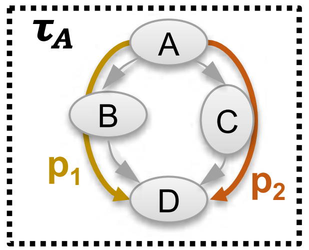
CL	RT	P
HI	5 ms	0
LO	7 ms	2
HI	3 ms	1

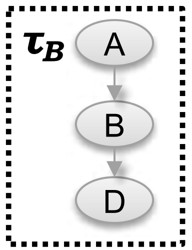
High criticality tasks require timing-guarantees

Time guarantees

Worst-Case Execution Time (WCET)

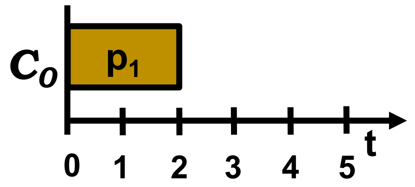
- Variations in the execution time

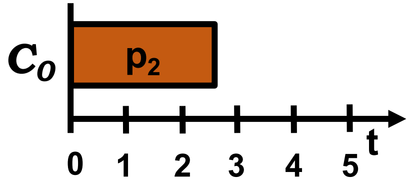




Application

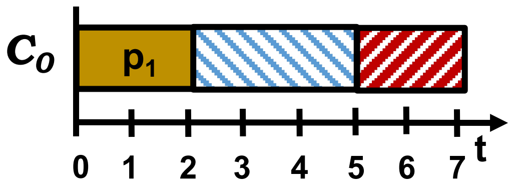
- Several execution paths 🟫🟧
- Data-dependent

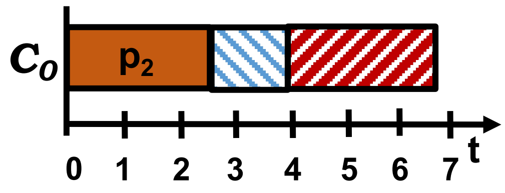


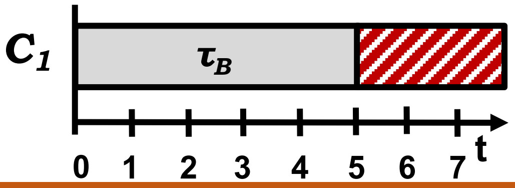


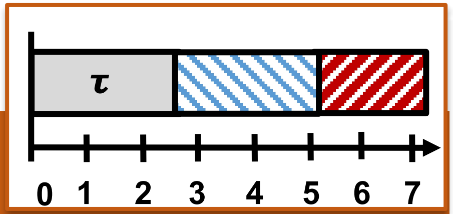
Platform

- Dynamic behavior ▨
  - Caches, branch predictors
- Shared resources ▨
  - Timing interference









Static WCET is safe, but pessimistic

WCET will (almost) never happen

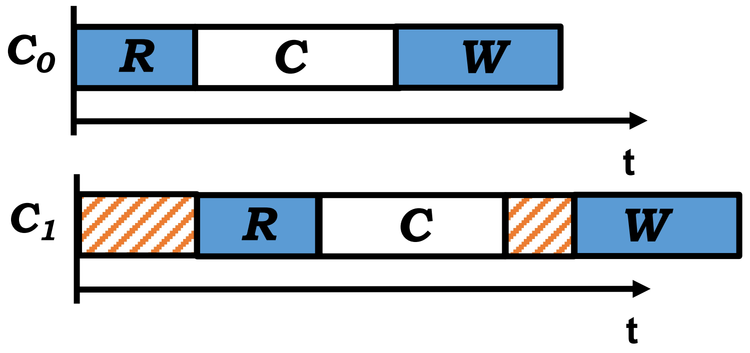
- **Actual execution is typically better than WCET estimation**
  - The actual execution path is not the worst
  - Some memory accesses were actually cache hits
  - Less interference occurred in shared resources

Can we reduce WCET pessimism ?

Reducing WCET pessimism

Typical approaches	Interference	Requirement
Isolation	Free	Maintain schedule

PREM
e.g. RTS'12



🟦 **Memory phase**

⬜ **Execution phase**

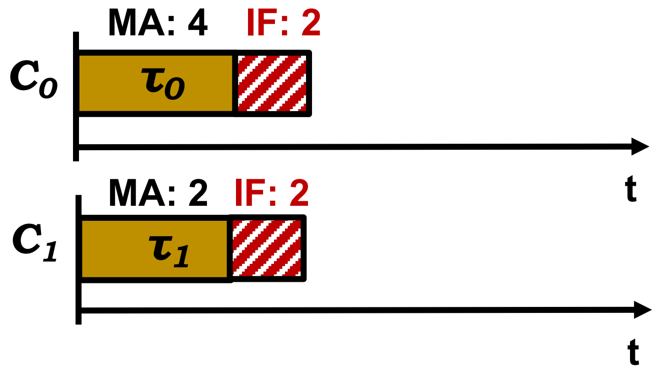
▨ **Idle**

G. Yao, R. Pellizzoni, S. Bak, E. Betti, and M. Caccamo, "Memory-centric scheduling for multicore hard real-time systems" Real-Time Systems, vol. 48, no. 6, pp. 681–715, 2012

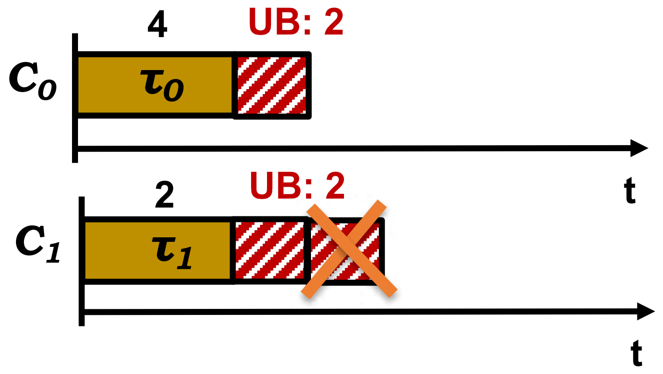
Reducing WCET pessimism

Typical approaches	Interference	Requirement
Isolation	Free	Maintain schedule
Interference-sensitive WCET	Controlled	Maintain schedule or bounds

Schedule
e.g. DATE'17



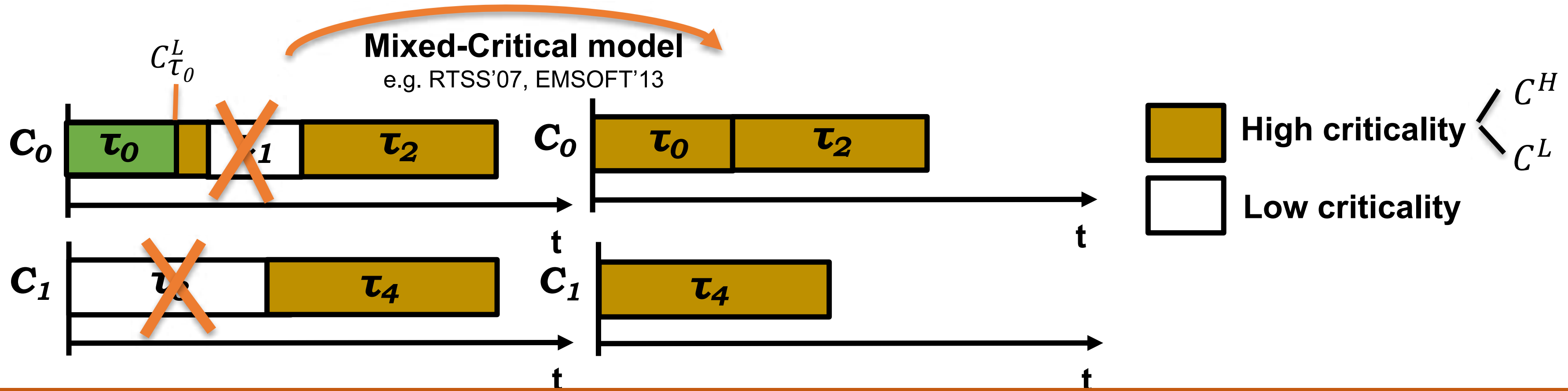
Bound resource usage
e.g. ECRTS'14



S. Skalistis and A. Simalatsar, "Near-optimal deployment of dataflow applications on many-core platforms with real-time guarantees", in DATE, 2017
J. Nowotsch, M. Paulitsch, D. Buhler, H. Theiling, S.Wegener, and M. Schmidt, "Multi-core Interference-Sensitive WCET Analysis Leveraging Runtime Resource Capacity Enforcement" in ECRTS, 2014

Reducing WCET pessimism

Typical approaches	Interference	Requirement
Isolation	Free	Maintain schedule
Interference-sensitive WCET	Controlled	Maintain schedule or bounds
Mode switch	Tolerant	Maintain bounds



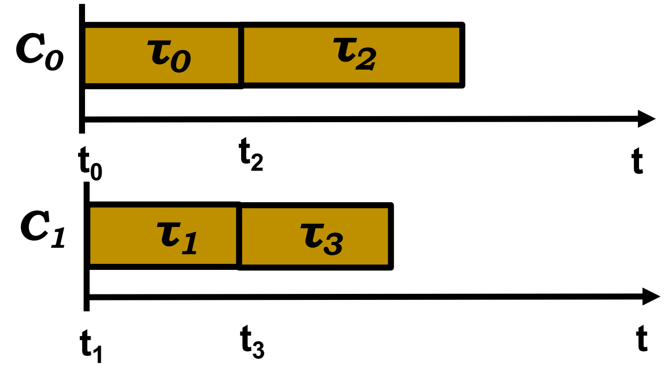
Requirements for safety impose limitations

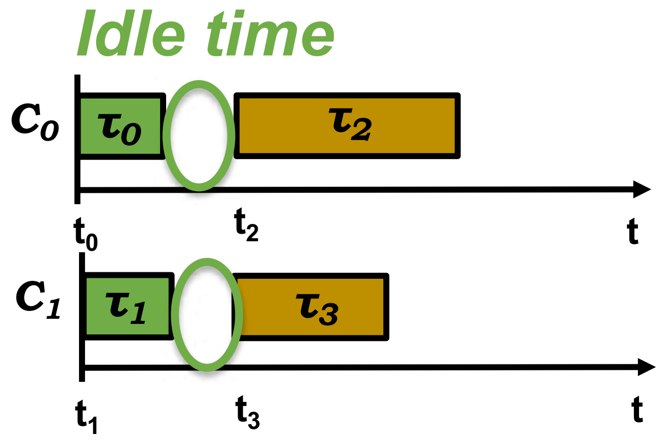
Maintaining schedule: Limitations

- **Time-triggered execution**
  - Fixed start time, computed statically

Core	Task	Time	Next
C_0	τ_0	t_0	τ_2
C_1	τ_1	t_1	τ_3
C_0	τ_2	t_2	-
C_1	τ_3	t_3	-

- **Cannot exploit: Early task termination**
  - Idle time

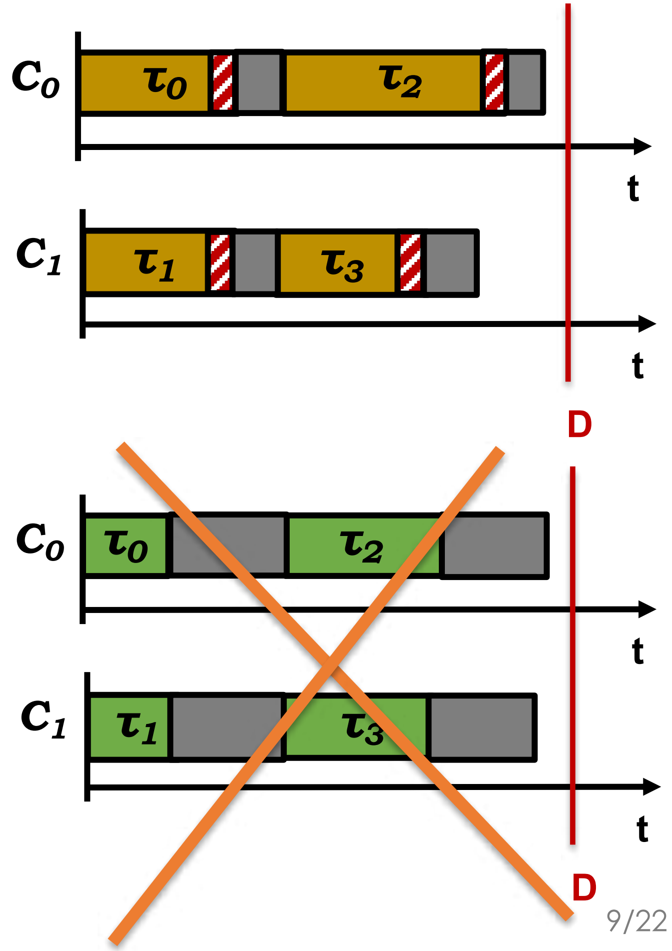




Run-time adaptation is required

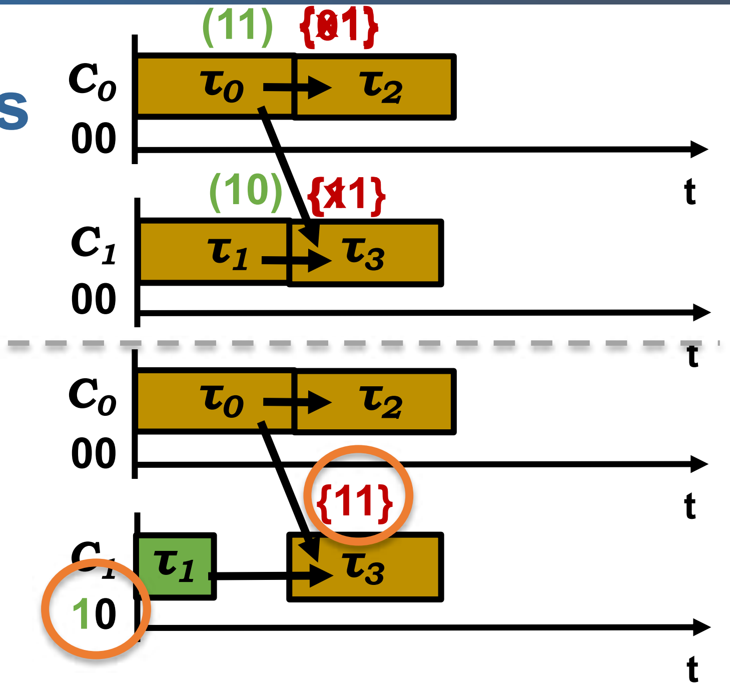
Run-time adaptation (RA)

- **Control mechanism** ■
  - Software
  - Hardware
- **Safe adaptation**
  - No deadline miss
  - No additional/Bounded interference ▨
- **Low overhead**
  - Not to negate adaptation gain



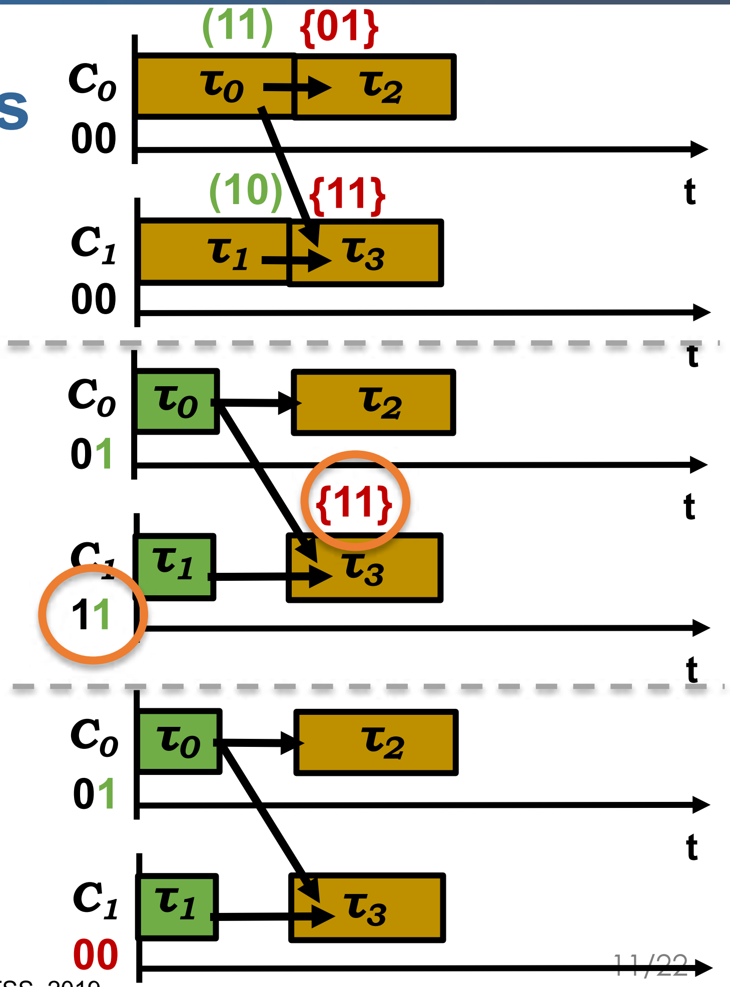
RA: isWCET schedule

- **Key idea: Preserve partial order of tasks**
- **Safe: No additional interference**
- **Implementation**
  - Insert scheduling dependencies
  - Encoded with bit vectors
    - Task: Notification, Ready
    - Core: Status



RA: isWCET schedule

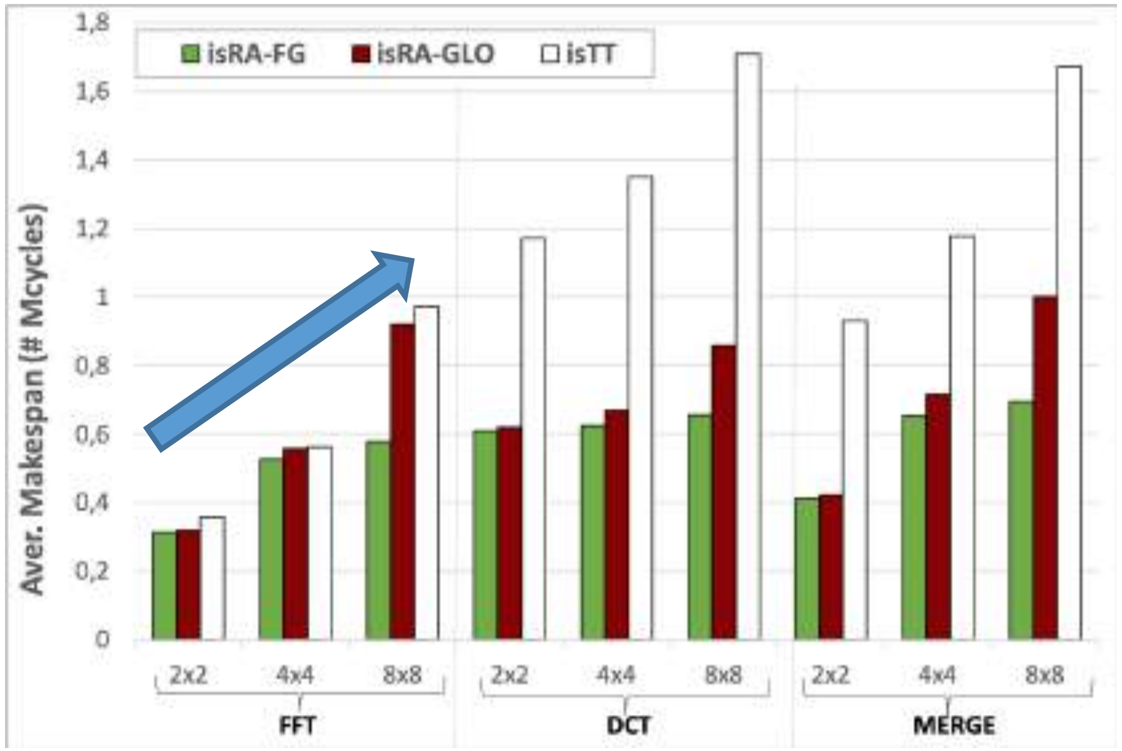
- **Key idea: Preserve partial order of tasks**
- **Safe: No additional interference**
- **Implementation**
  - Insert scheduling dependencies
  - Encoded with bit vectors
    - Task: Notification, Ready
    - Core: Status
  - Concurrency: Status
    - Protection mechanisms



Evaluation

- **Methods:**
  - isTT: Time-triggered
  - isRA: Run-time Adaptation
    - GLO: Global protection mechanism
    - FG: Distributed protection mechanism
- **TI TMS3206678**
  - 8 DSPs @ 1GHz

>50%



Can we do better?

Exploit run-time information

- **Available only during actual execution**
  - Execution progress
  - Current state of hardware components
- **Dynamically improve bounds computed statically**
  - Allowed interference from co-runner tasks
  - Upper bounds in resource usage
  - WCET estimations

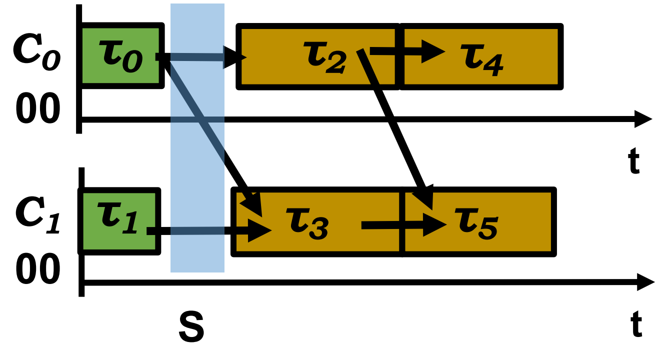
Exploit run-time information

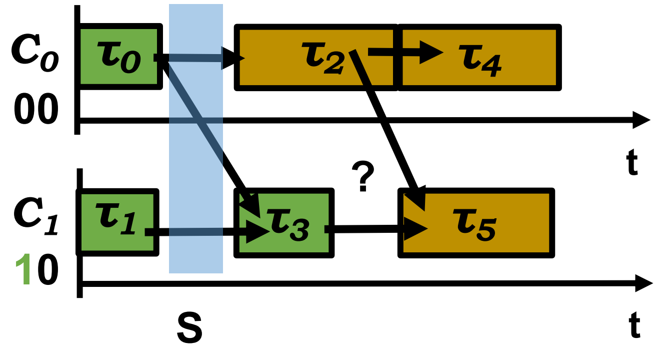
- **Available only during actual execution**
  - **Execution progress**
  - Current state of hardware components

- **Dynamically improve bounds computed statically**
  - **Allowed interference from co-runner tasks**
  - Upper bounds in resource usage
  - **WCET estimations**

Progress: isWCET schedule

- Key idea: Relax partial order of tasks
- Safe: If extra interference is sustained
- Implementation
  - As before: Bit vectors (sch. dependencies)
  - Time Slack: Minimum speed-up in task execution among all cores
  - Relax: WCET of extra interferences ≤ Slack
    - Remove scheduling dependencies

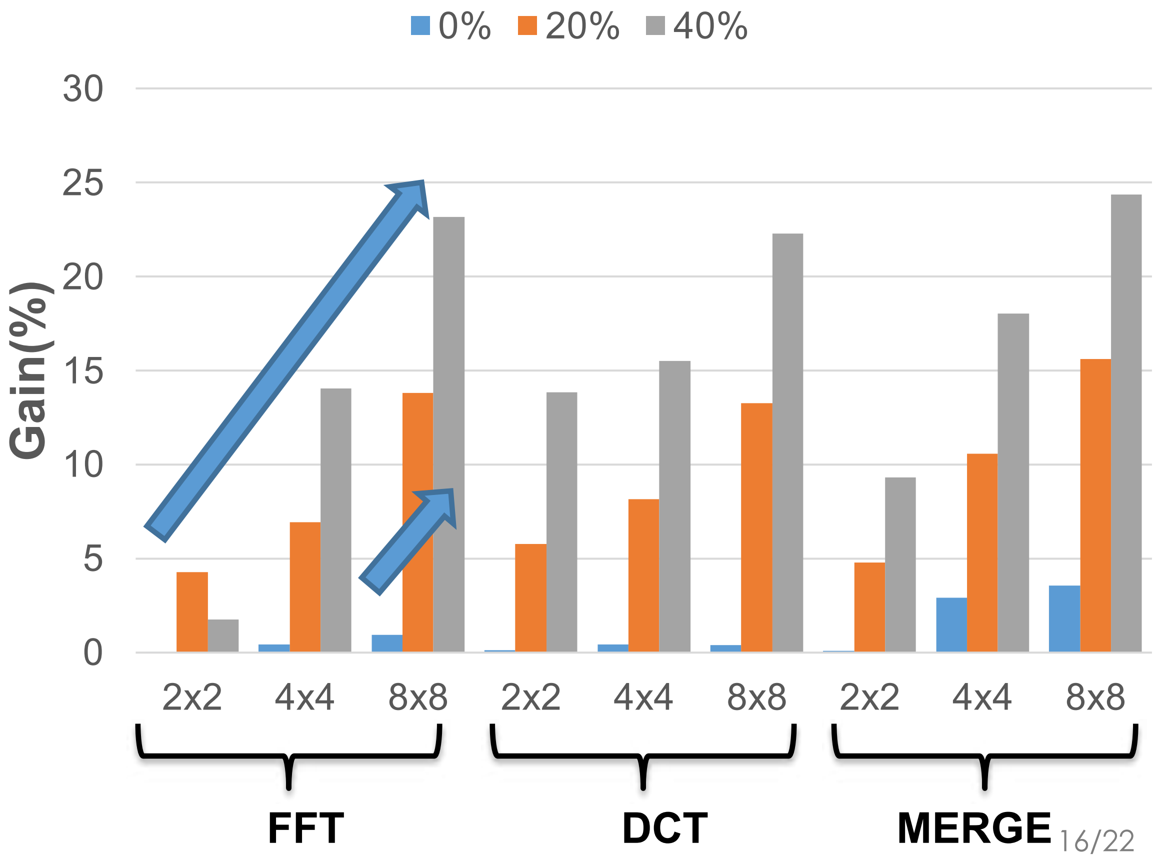




S. Skalistis and A. Kritikakou, "Dynamic Interference-Sensitive Run-time Adaptation of Time-Triggered Schedules", in ECRTS, 2020

Evaluation

- **Methods**
  - isRA-FG
  - isRA-DYN
- **Timing Variability (≤70%)**
  - Paths
  - Cache
- **TI TMS3206678**
  - 8 DSPs @ 1GHz



Progress: WCET estimation

- **Key idea: Compute the Remaining WCET (RWCET)**
  - WCET of the code that has not executed yet
- **Safe: Removing WCET of executed part**
- **Implementation**
  - Insert monitoring points
  - Compute partial WCET
    - Static analysis
    - Measurement-based
  - Compute RWCET based on partial WCET

Instrument source code

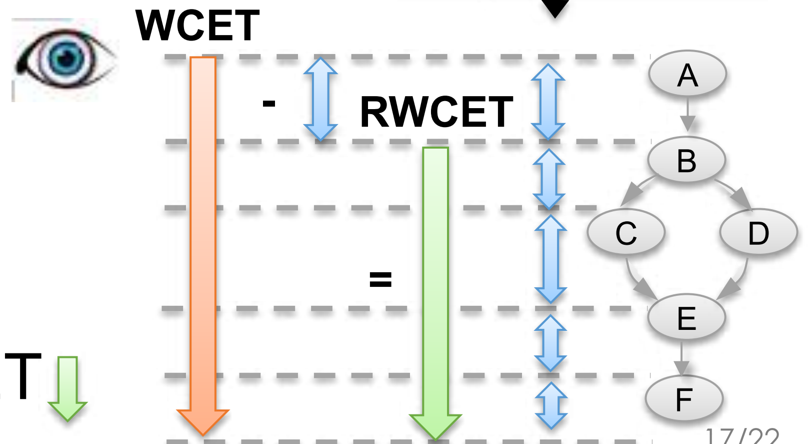
```
even = 0;
odd = 0;
If (iso==0) RTC(a);
for (i=0 ; i<N ; i++) {
   if (i%2 == 0)
      even++;
   else
      odd++;
```

compiler

Assembly

```
add r3,r0,r0 …
```

Graph construction



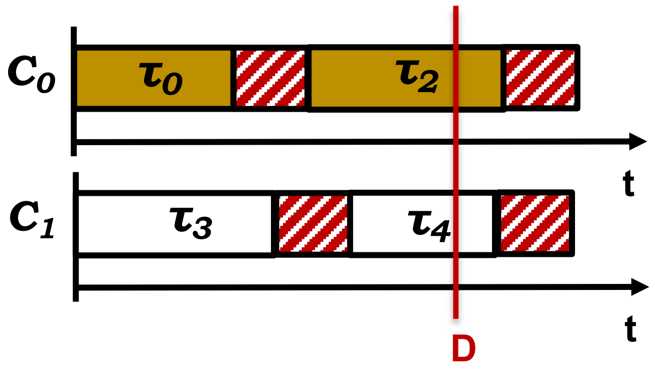
A. Kritikakou, C. Pagetti, O. Baldellon, M. Roy, C. Rochange, “Run-Time Control to Increase Task Parallelism In Mixed-Critical Systems”, in ECRTS’14

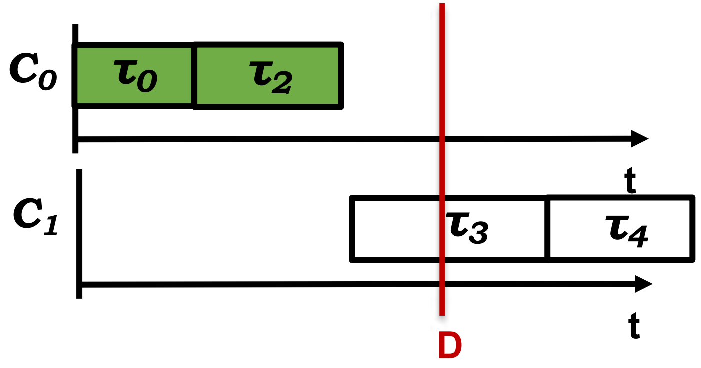
RWCET: Mode switch

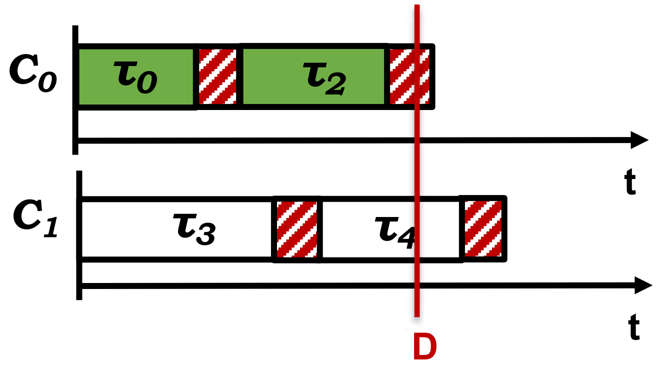
- **Max load mode:**
  - HC tasks 🟫
  - LC tasks ⬜
    - Interference
    - WCET>D

- **Isolation mode:**
  - Only HC tasks
  - If time, LC tasks

- **Mode switch:**
  - Max load mode
  - If risk for HC tasks
    - Isolation mode

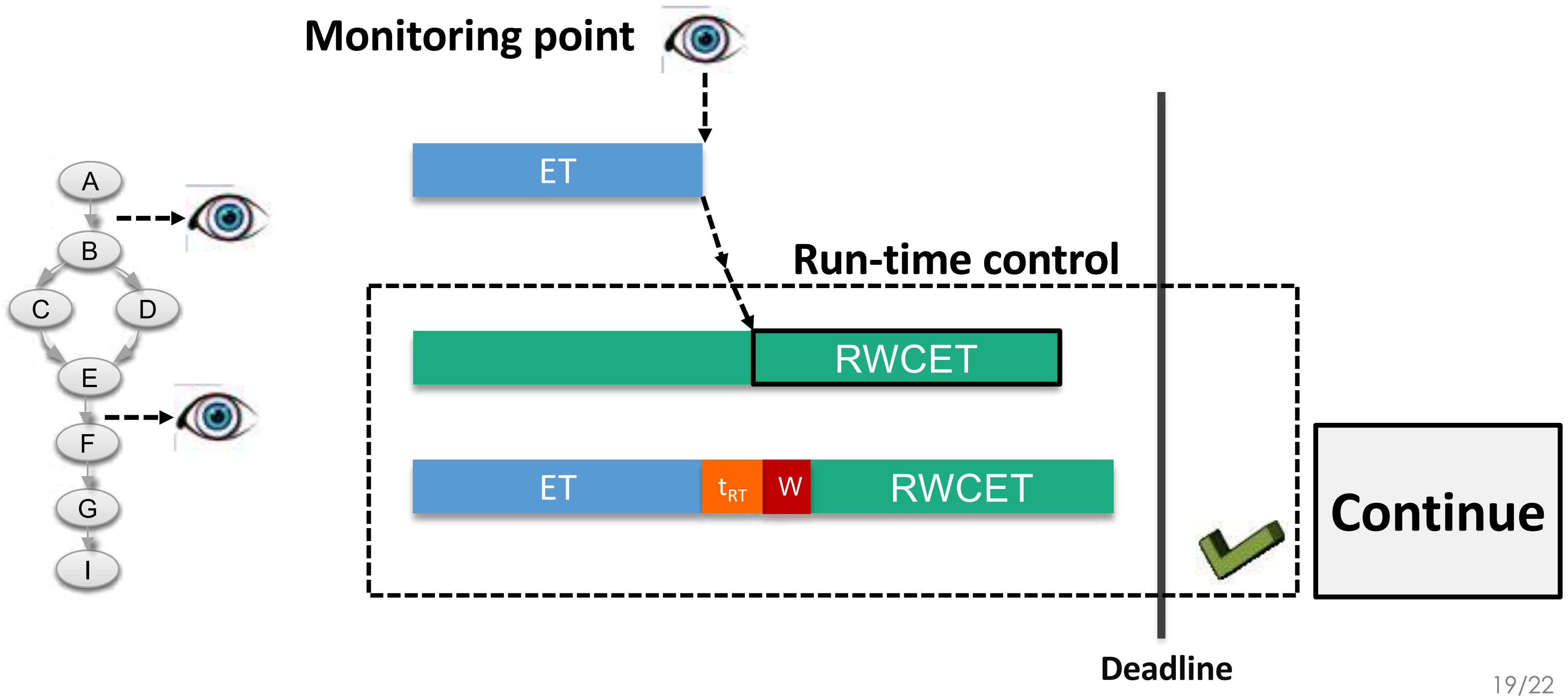




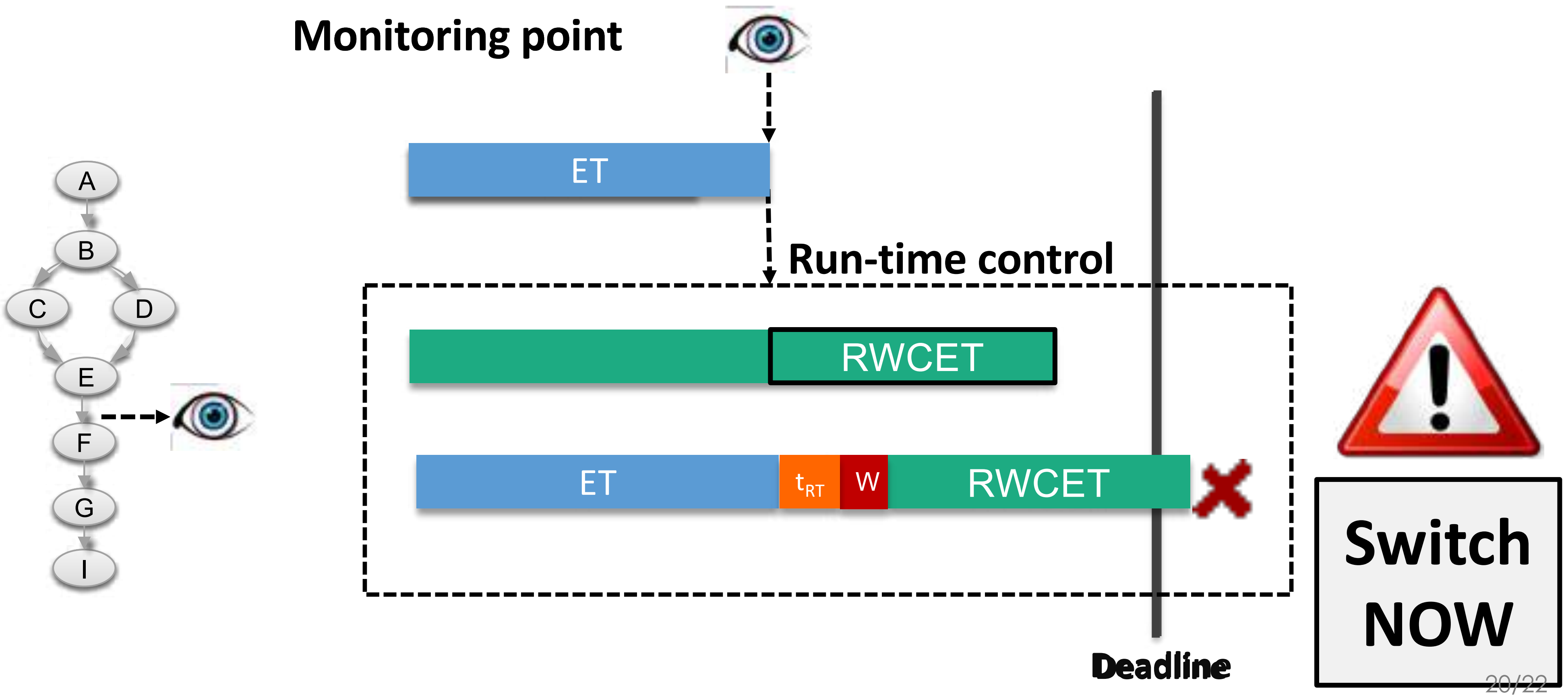


Safety condition

RWCET: Safety condition

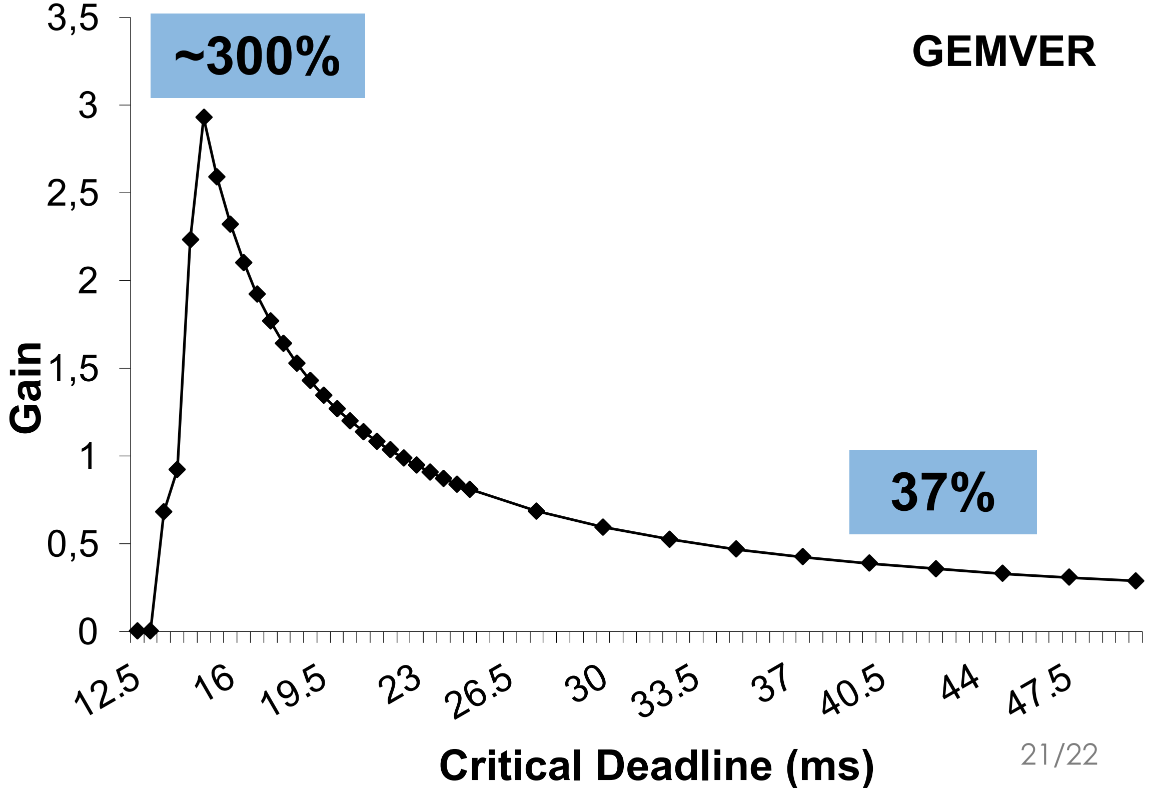


RWCET: Safety condition



Results

- **Methods:**
  - isolation: Only HC tasks, if time LC tasks
  - Mode-switch: RWCET
- **Workload:**
  - 2 cores: HC tasks
  - 6 cores: LC tasks
- **TI TMS3206678**
  - 8 DSPs @ 1GHz



Conclusions & Further opportunities

- **WCET pessimism**
- **Run-time adaptation approaches:**
  - Execution progress
    - Interference-sensitive schedule
    - WCET estimation
  - Software
- **Hardware mechanisms for run-time adaptation**
- **Approaches**
  - Take into account state of hardware components
  - Combine with scheduling techniques

Thank You

Questions?

In any case, feel free to contact me:
angeliki.kritikakou@irisa.fr