ECRTS 2014
8-11 July
Madrid, Spain

Analysis Techniques for Supporting Harmonic Real-Time Tasks with Suspensions

ECRTS'14, Madrid, Spain

July 10th, 2014

Cong Liu, Jian-Jia Chen, Liang He, Yu Gu

Suspensions are Pervasive in Practice

When accessing external devices, the task is suspended (blocked) by OS on CPU

Negative Impact due to Suspensions on Hard Real-Time (HRT) Schedulability

Uniprocessor scheduling of tasks that may suspend for at most one time is NP-hard in the strong sense

On a uniprocessor, algorithms such as earliest-deadline first are not optimal even with a k-speed processor

F. Ridouard, P. Richard, and F. Cottet. "Negative results for scheduling independent hard real-time tasks with self-suspensions." In Proc. of the 25th RTSS, pp. 47-56, 2004.

Harmonic Suspending Task Systems

- A special case of practical relevance
  - Harmonic periods seen in many settings, e.g., avionics
- Harmonic periods yield better schedulability for ordinary periodic task systems

Harmonic Suspending Task Systems

- A special case of practical relevance
  - Harmonic periods seen in many settings, e.g., avionics
- Harmonic periods yield better schedulability for ordinary periodic task systems

This paper:

Can we derive better *uniprocessor* and *multiprocessor* schedulability tests for HRT periodic suspending tasks systems with harmonic periods?

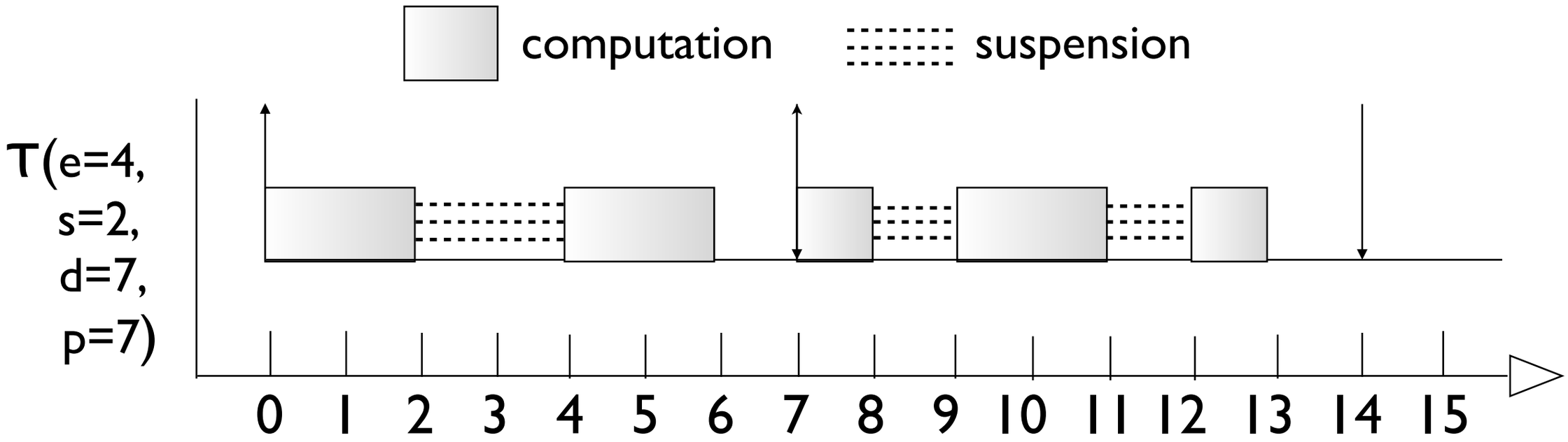
Our Suspending Task Model

Implicit-deadline periodic (synchronous release)
suspending task: $\tau(e, s, d, p)$

e: worst-case execution time
s: worst-case suspension length
d: relative deadline
p: period

Our Suspending Task Model

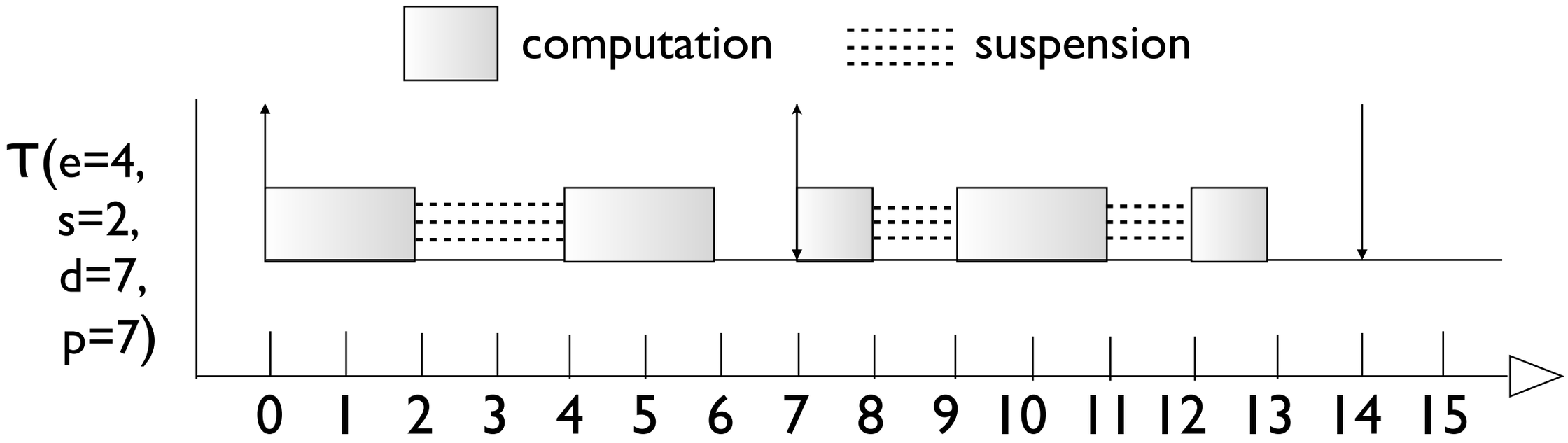
Implicit-deadline periodic (synchronous release) suspending task: $\tau(e, s, d, p)$



Jobs may alternate between computation and suspension phases without any restriction on how they interleave

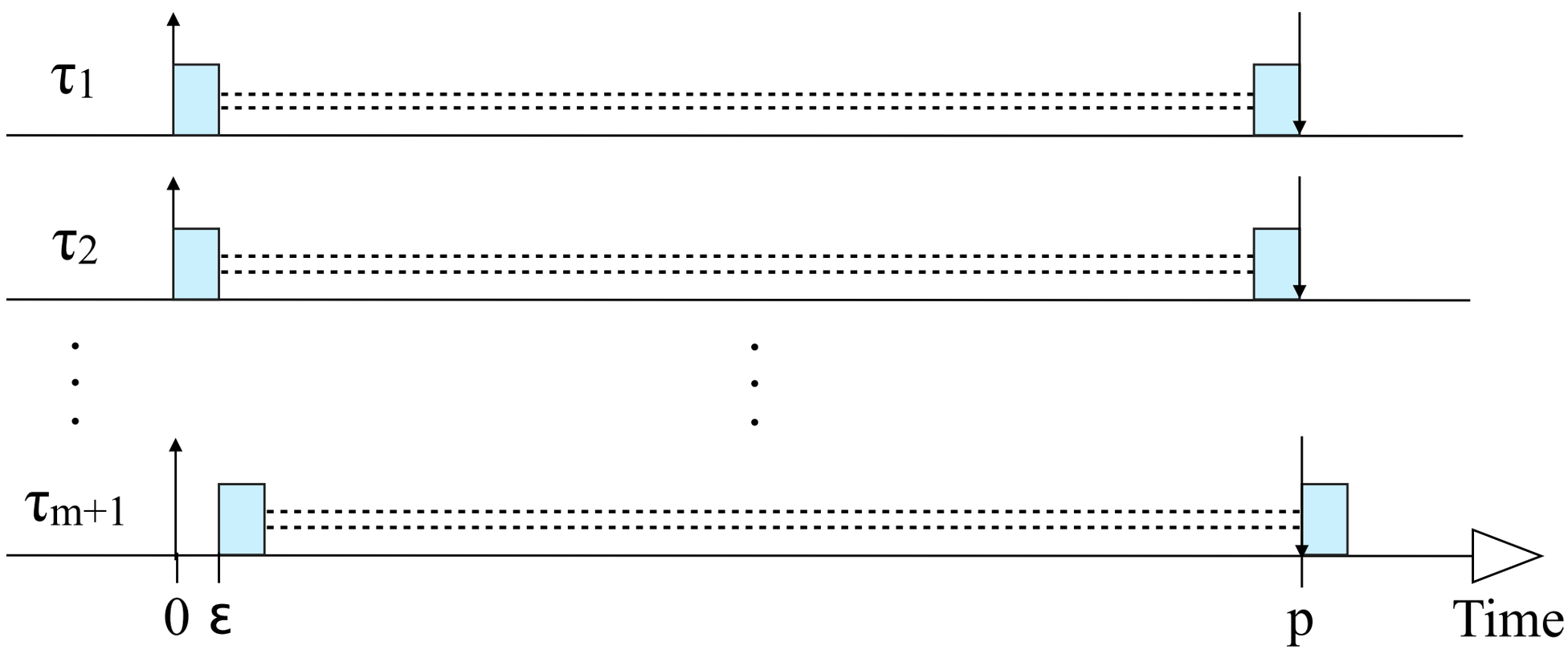
Our Suspending Task Model

Implicit-deadline periodic (synchronous release) suspending task: τ(e, s, d, p)



- Task periods are pairwise divisible
- Rate monotonic scheduling

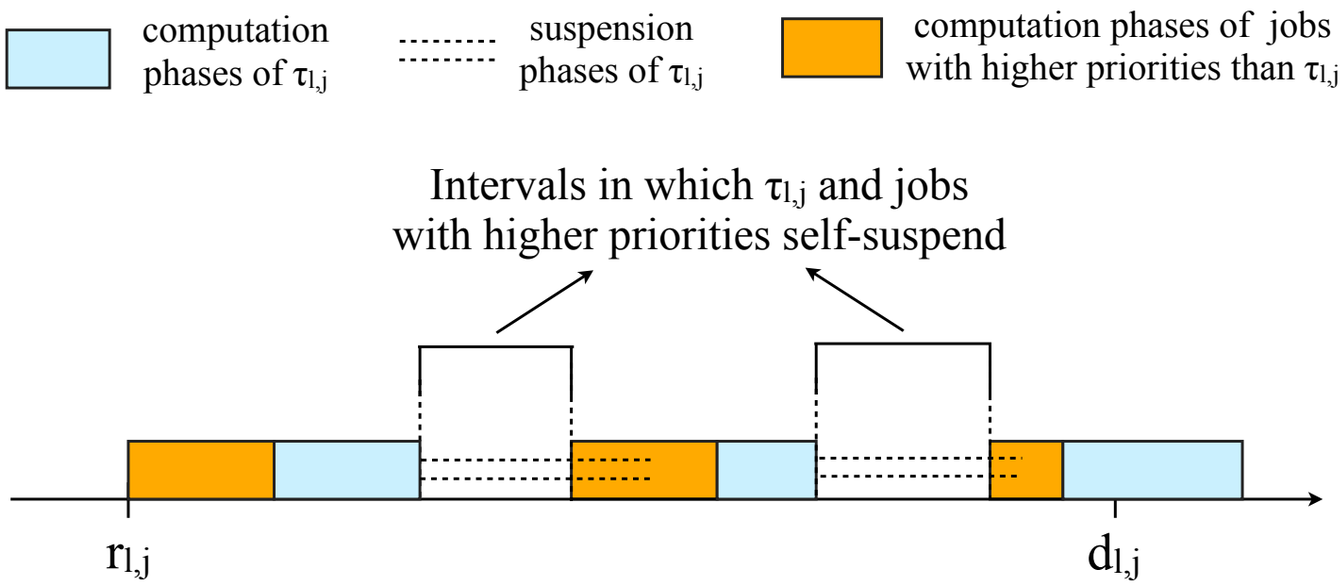
Worst-Case Behavior due to Suspensions



- Suspensions reduce the time available for completion
- Non-deterministic suspension patter
  - Worst case: when one task suspends, all tasks may suspend at the same time!

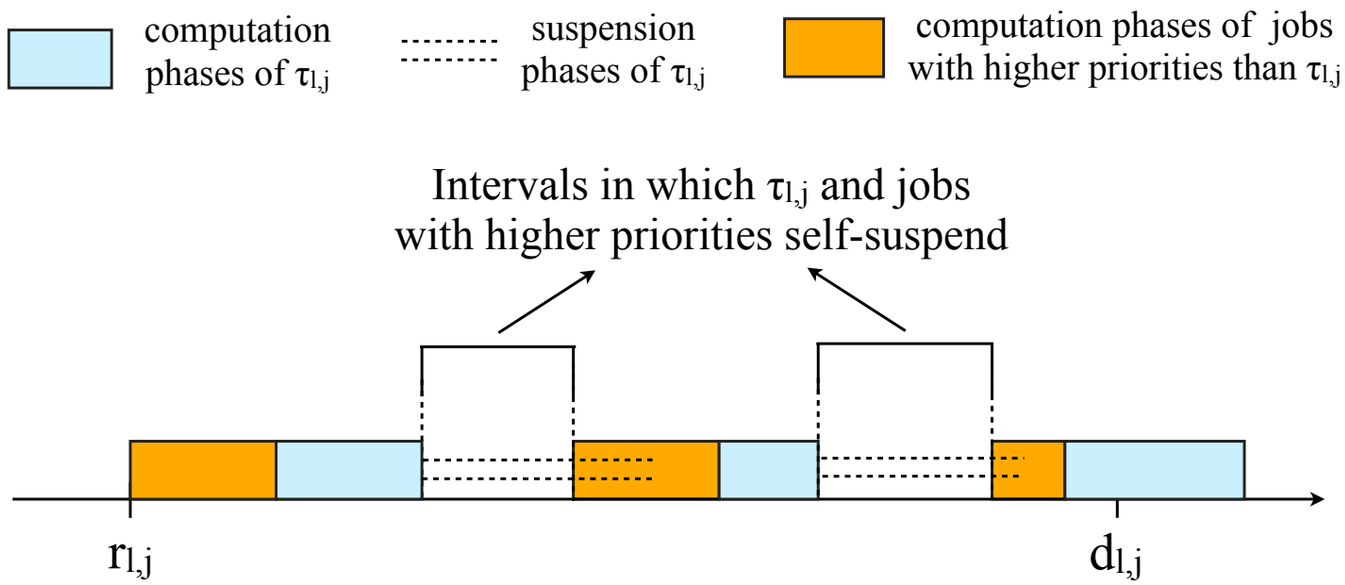
Uniprocessor Schedulability Analysis

- Let be $\tau_{l,j}$ the first job that misses its deadline
  - Objective: determine conditions necessary for this to happen



Uniprocessor Schedulability Analysis

- Let be $\tau_{l,j}$ the first job that misses its deadline
  - Objective: determine conditions necessary for this to happen



Avoid the worst case:

All suspending tasks may have jobs suspending at idle instants: the total work in $[r_{l,j}, d_{l,j})$ does not need to exceed p_l in order for $\tau_{l,j}$ to miss its deadline

Key Observation

- $\tau_{l,j}$ must suspend at any idle instant within $[r_{l,j}, d_{l,j})$
  - Since it misses deadline

Key Observation

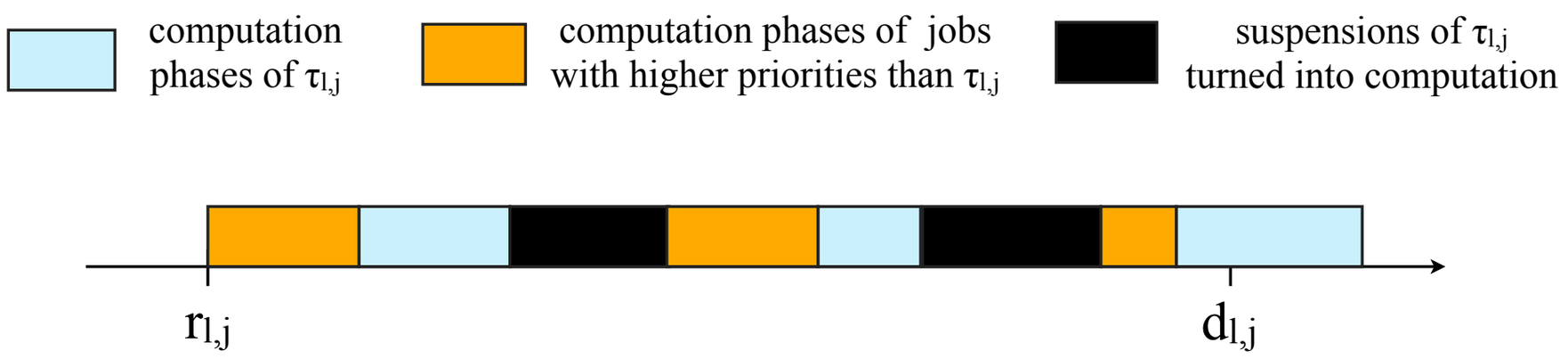
- $\tau_{l,j}$ must suspend at any idle instant within $[r_{l,j}, d_{l,j})$
  - Since it misses deadline

Treating $\tau_{l,j}$'s suspensions within such idle intervals as computation forces $[r_{l,j}, d_{l,j})$ to be a busy interval, at the cost of treating only one task's suspensions as computation

Key Observation

- $\tau_{l,j}$ must suspend at any idle instant within $[r_{l,j}, d_{l,j})$
  - Since it misses deadline

Treating $\tau_{l,j}$'s suspensions within such idle intervals as computation forces $[r_{l,j}, d_{l,j})$ to be a busy interval, at the cost of treating only one task's suspensions as computation



Uniprocessor Schedulability Test: Θ(1) suspension-related utilization loss

- An HRT synchronous periodic harmonic suspending task system with total utilization ≤ 1 is schedulable under RM on a uniprocessor if:

$$\max_k \left\{ \sum_{i=1}^{k} u_i + \frac{s_k}{p_k} \right\} \leq 1$$

Suspending Task Partitioning on Multiprocessors

$$\max_k \left\{ \sum_{i=1}^{k} u_i + \frac{s_k}{p_k} \right\} \leq 1$$

Utilization loss due to this term is caused by only one such task

Suspending Task Partitioning on Multiprocessors

$$\max_k \left\{ \sum_{i=1}^{k} u_i + \frac{s_k}{p_k} \right\} \leq 1$$

Utilization loss due to this term is caused by only one such task

Intuition: assign tasks with large suspension ratios to the same processor

Example Task Partition

- A two-processor suspending task system with six tasks: $\tau_1(1, 4, 5)$, $\tau_2(3, 5, 10)$, $\tau_3(2, 4, 10)$, $\tau_4(1, 2, 5)$, $\tau_5(12, 0, 20)$, $\tau_6(10, 0, 20)$
  - Has a total utilization of two

Example Task Partition

- A two-processor suspending task system with six tasks: $\tau_1(1, 4, 5)$, $\tau_2(3, 5, 10)$, $\tau_3(2, 4, 10)$, $\tau_4(1, 2, 5)$, $\tau_5(12, 0, 20)$, $\tau_6(10, 0, 20)$
  - Has a total utilization of two
- Successfully partitioned under our algorithm
  - Processor 1: τ_1, τ_2, τ_6
  - Processor 1: τ_3, τ_4, τ_5

$$\max_k \left\{ \sum_{i=1}^{k} u_i + \frac{s_k}{p_k} \right\} \leq 1$$

- Because τ_1 and τ_2 that are the two tasks with the largest suspension to period ratios are assigned to the same processor
- Thus, this ratio of τ_1 (which is 0.8!) is "masked" by τ_2

Multiprocessor Schedulability Test: Θ(m) suspension-related utilization loss

- A synchronous periodic harmonic suspending task system is schedulable on m processors if

$$U_{sum} \leq m - U_{m-1} - V_m$$

- where U_{m-1} denotes the sum of m-1 largest task utilizations, and V_m denotes the sum of m largest task suspension-to-period ratios

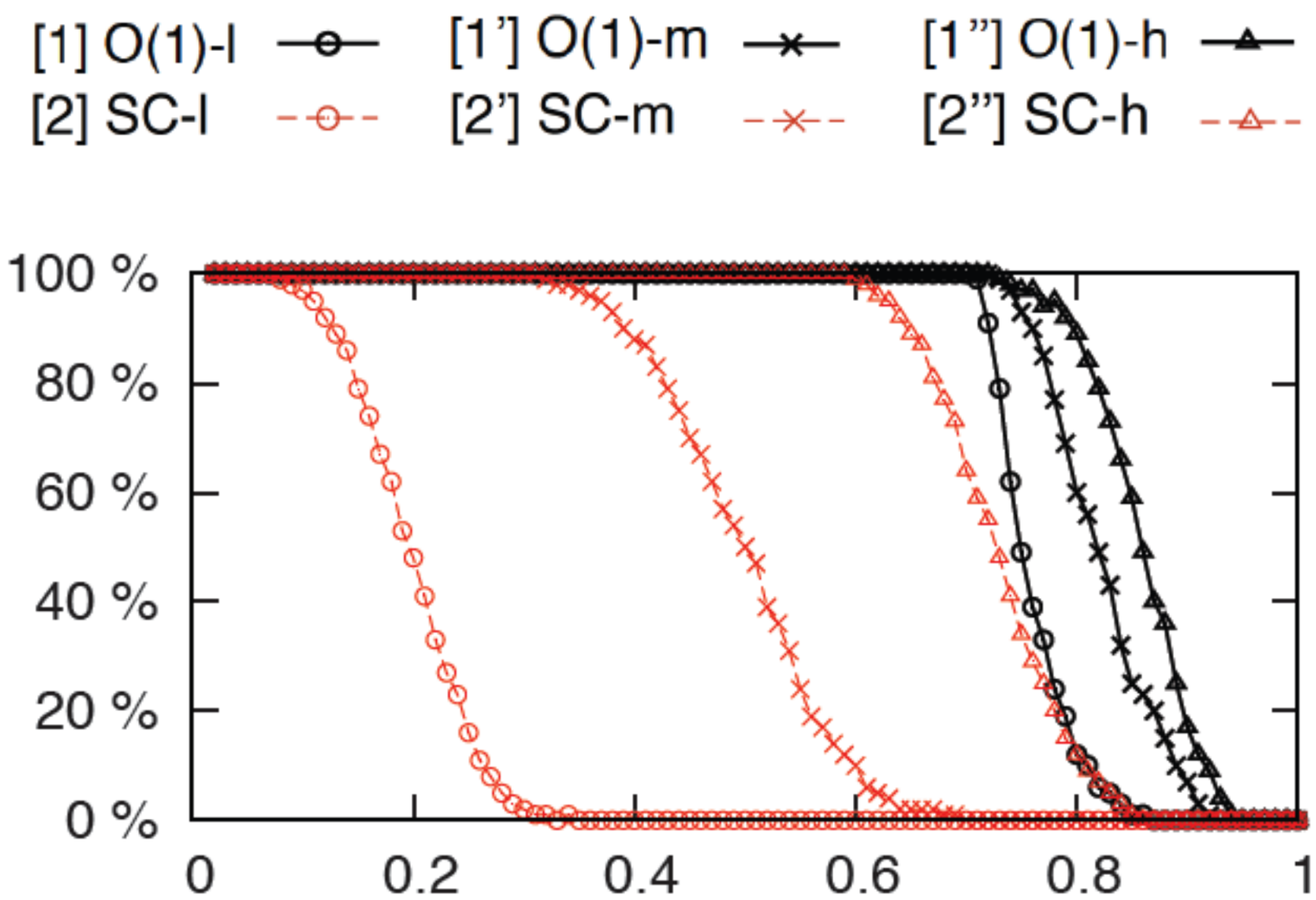
Experiments

- Harmonic task periods:[2ms, 1024ms]
- Task utilizations:
  - [0.005, 0.1] (light)
  - [0.1, 0.3] (medium)
  - [0.3, 0.5] (heavy)
- Suspension lengths:
  - $[0.005(1-u_i)p_i, 0.1(1-u_i)p_i]$ (short)
  - $[0.005(1-u_i)p_i, 0.1(1-u_i)p_i]$ (medium)
  - $[0.005(1-u_i)p_i, 0.1(1-u_i)p_i]$ (long)
- Three cases: m = 1, 4, 8
- 10, 000 task set per experiment

Experiments

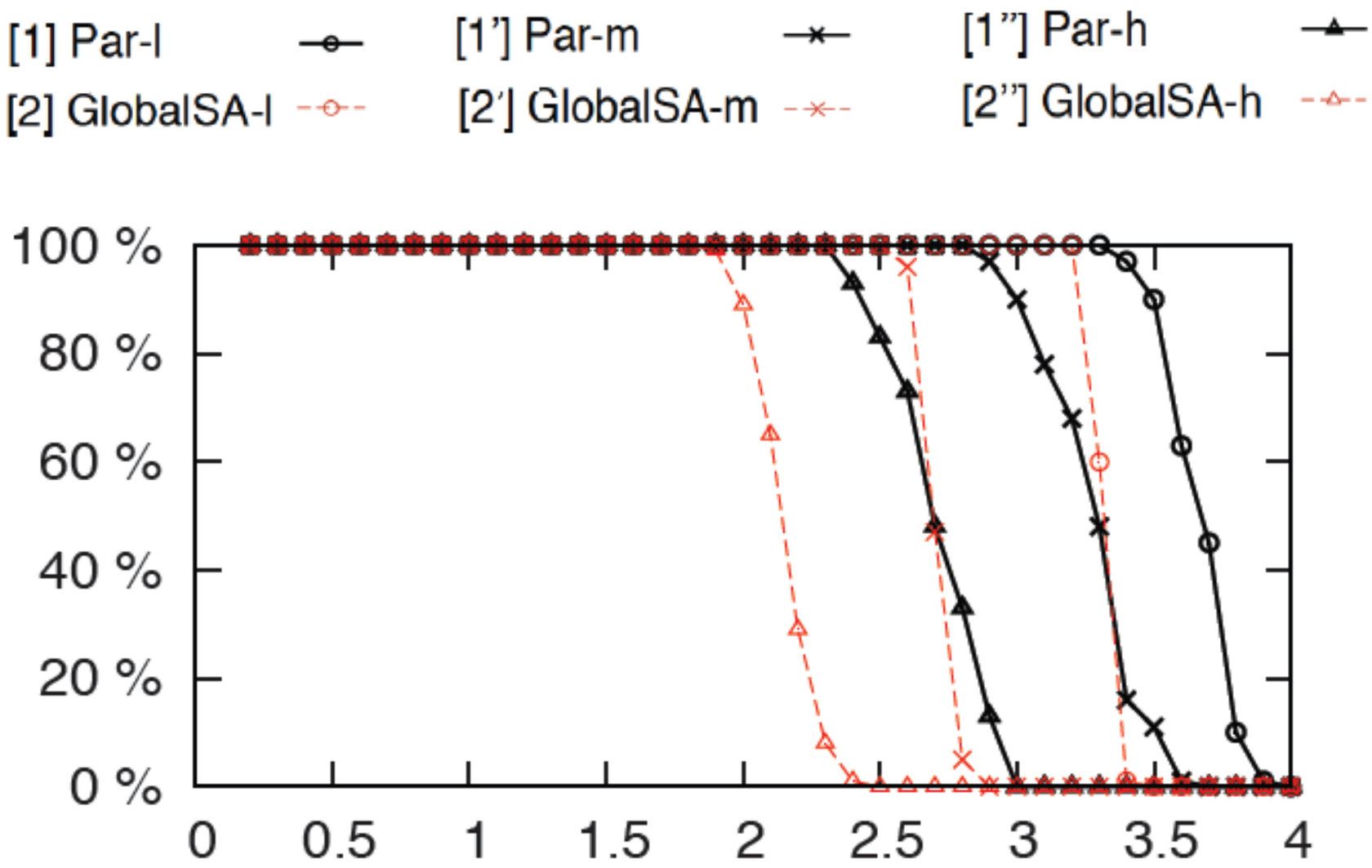
- Uniprocessor
  - Against the suspension-oblivioius approach "SC" which converts all tasks' suspensions into computation
  - Under SC, schedulable if the total utilization of the transformed task system is ≤ 1
- Multiprocessor
  - Against the only existing multiprocessor suspension-aware schedulability test "GlobalSA" under global scheduling

Uniprocessor Results



(b) moderate suspension length

Multiprocessor Results



(a) m = 4; short suspension length

Conclusion and Future Work

- New schedulability analysis for <u>periodic suspending task systems with harmonic periods</u>
  - Uniprocessor: Θ(1) suspension-related utilization loss
  - Multiprocessor: Θ(m) suspension-related utilization loss under a partition-based approach
- Experiments demonstrate the effectiveness

Why Other Settings Do NOT Work?

- Sporadic releases?
  - Non-deterministic releasing pattern ☛ unknown worst-case behavior (no known corresponding critical instant theorem)
- Arbitrary periods?
  - Similar issue: non-deterministic suspension pattern ☛ no known bound on the amount of high-priority work during a specific interval
- EDF scheduling?
  - Suspensions ☛ invalid property of the "idle instant"

Promising Current and Future Work: Bursty-Interference Analysis

- How to deal with the nondeterministic execution behaviors due to the general suspension pattern?
- How to accurately analyze the worst-case amount of interference due to higher-priority suspending tasks during certain intervals?

Promising Current and Future Work: Bursty-Interference Analysis

- How to deal with the nondeterministic execution behaviors due to the general suspension pattern?
- How to accurately analyze the worst-case amount of interference due to higher-priority suspending tasks during certain intervals?

Thanks!

Questions?