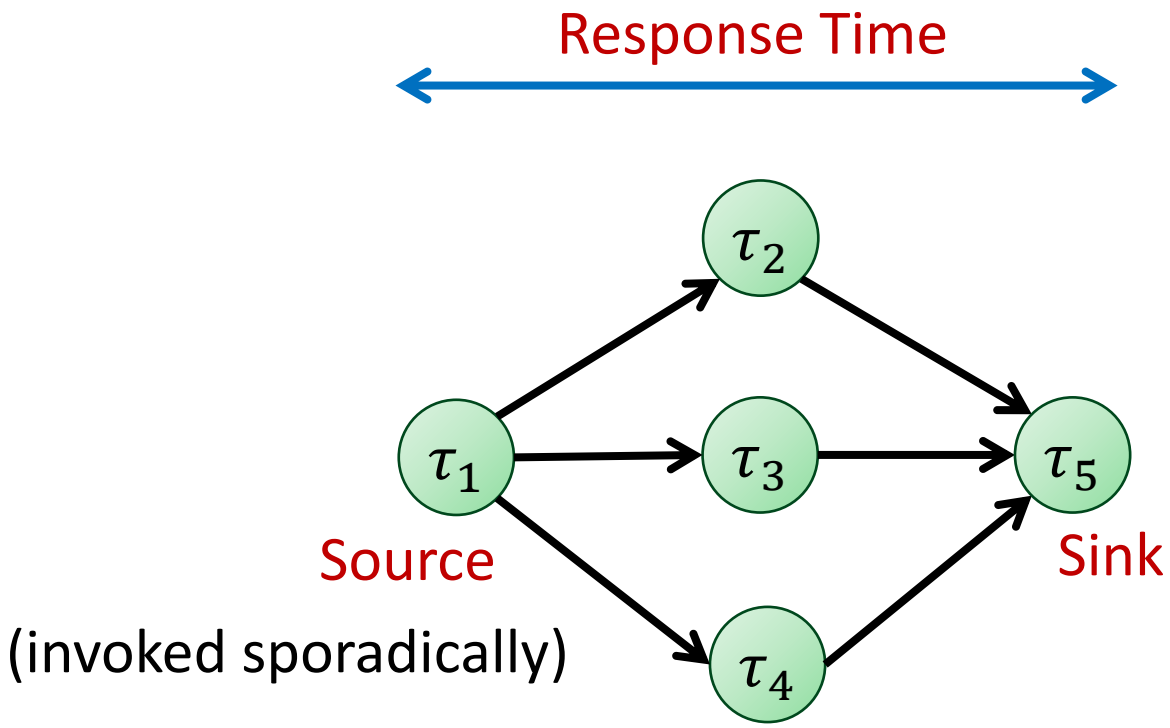
A Soft-Real-Time Optimal Scheduler for DAG Tasks with Node-Level Self Dependencies

Shareef Ahmed and James H. Anderson

UNIVERSITY of SOUTH FLORIDA

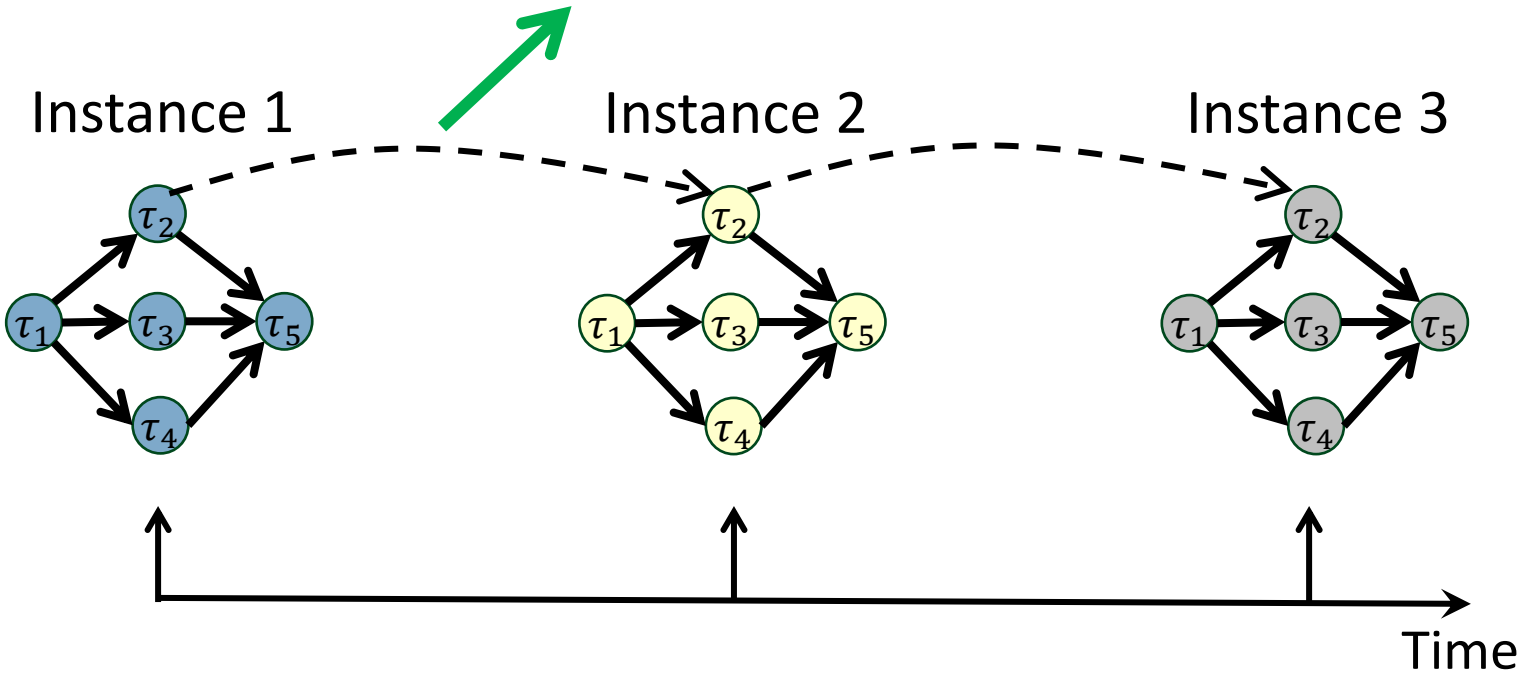
Processing Graph

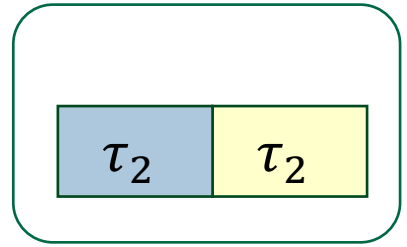


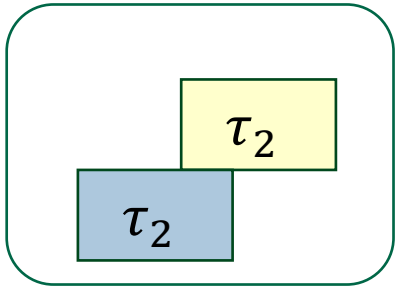
- Node = Task
- Edge = Precedence constraint
- Goal:
  - Bounded response time (SRT)
  - Response time can exceed period

Self-Dependencies Among Node Instances

τ_2 must complete before τ_2 can start execution (Example: Object tracking)



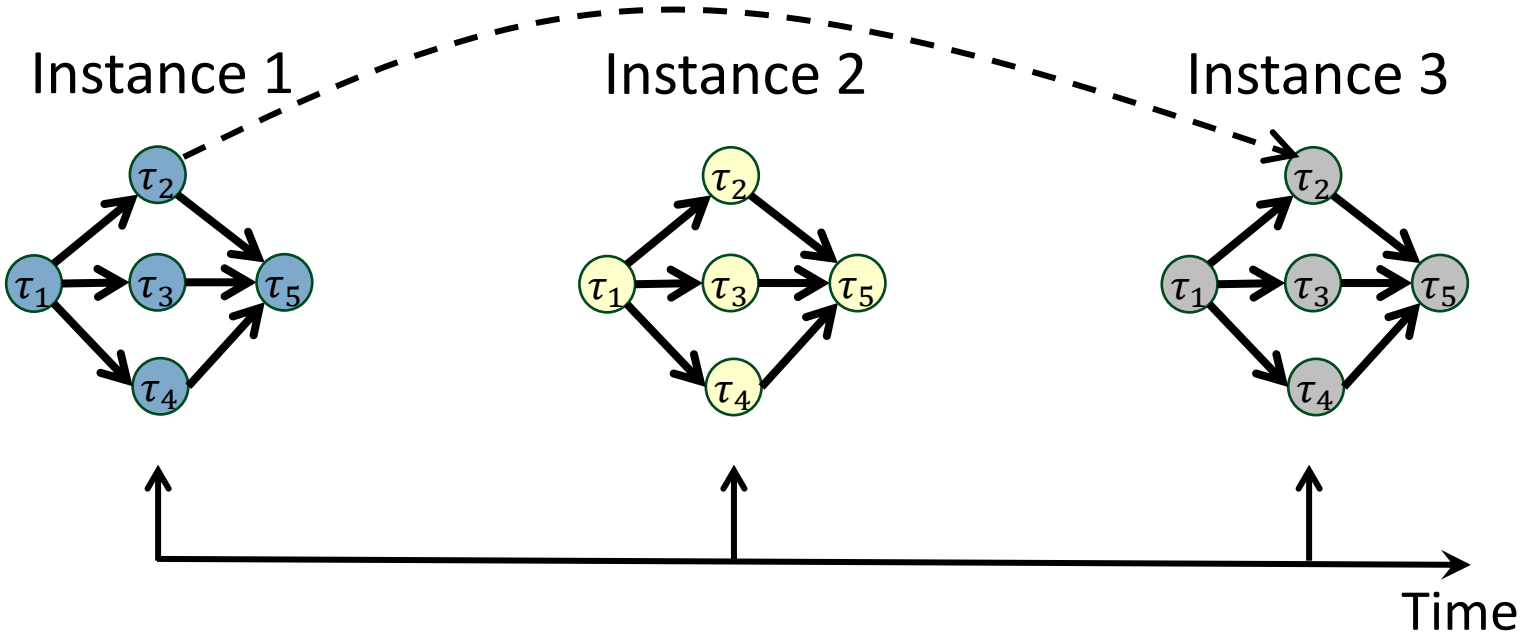




Dependency level = Parallelism level = 1

Self-Dependencies Among Node Instances

Dependency/parallelism level can be arbitrary
[rp-sporadic task model, Amert et al. RTSS 2020]



Dependency level = Parallelism level = 2

Feasibility And Optimality

SRT-Feasible Systems

Total utilization ≤ number of processor and Task utilization ≤ Dependency level

[Amert et al., RTSS 2020]

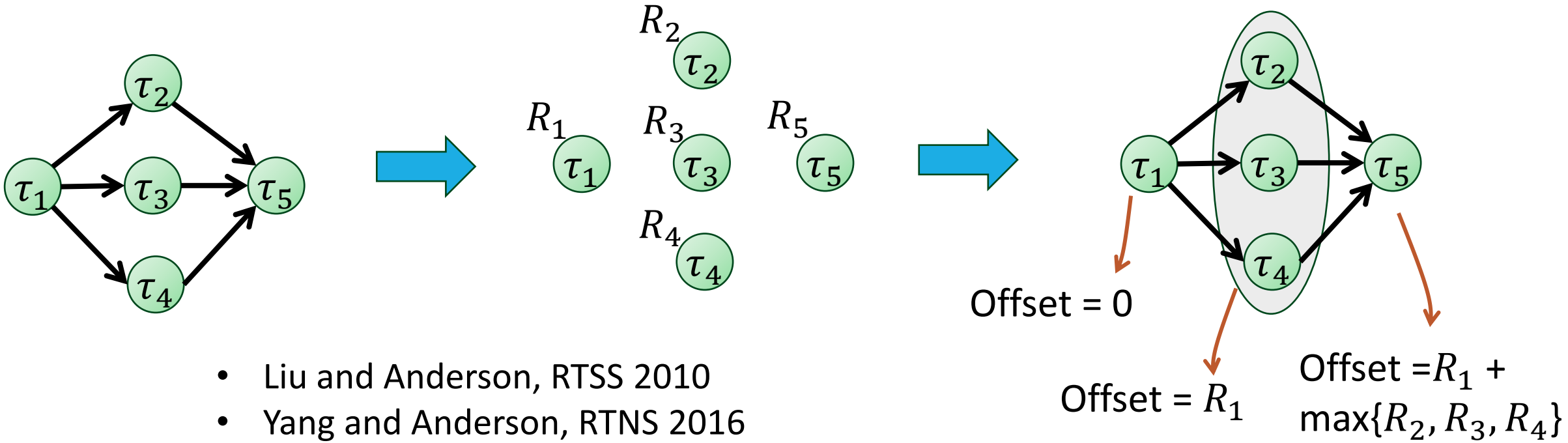
SRT-Optimal Scheduler

Bounded response time for any SRT-feasible system

SRT-Optimal Scheduler: Offset-Based Scheduling

SRT-optimal but R_i values can be pessimistically large

R_i = Response-time bound
(assuming independent tasks)

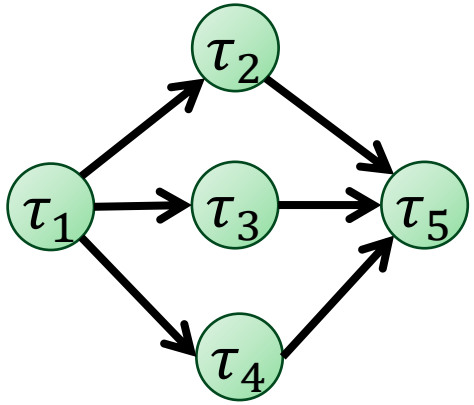


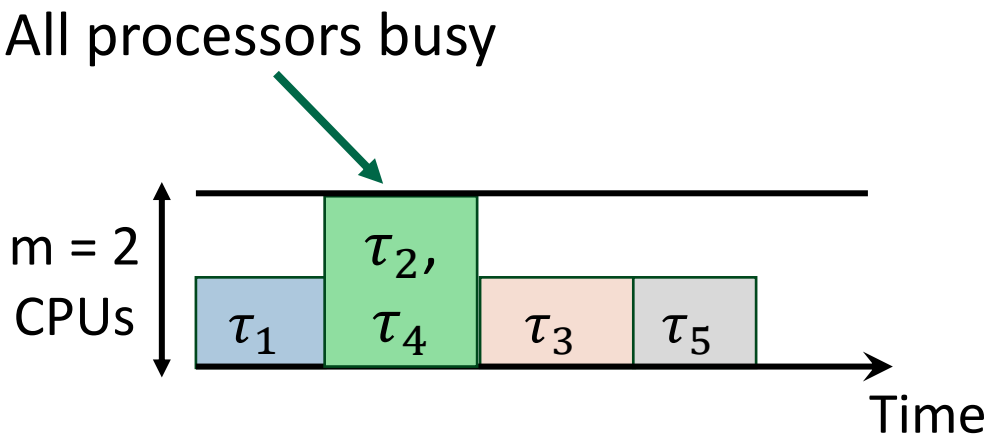
- Liu and Anderson, RTSS 2010
- Yang and Anderson, RTNS 2016
- Amert et al., RTSS 2020

Non-Decomposition-Based Analysis

Applicable for HRT DAG tasks

- Graham, Siam J. of Appl. Math., 1969
- Melani et al., ECRTS 2015
- He et al., RTSS 2022
- Fonseca et al., RTJ 2019

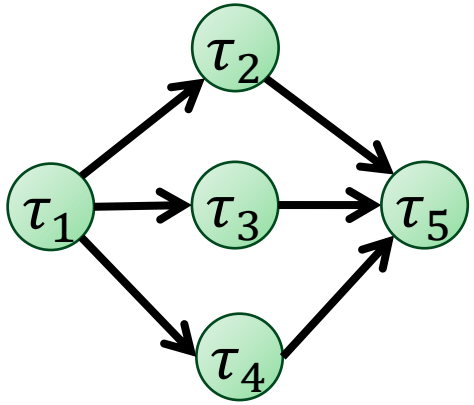


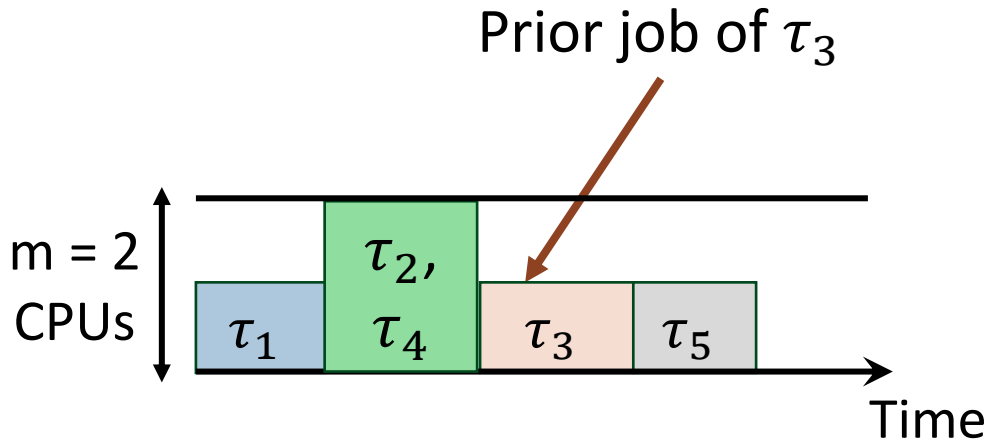


Non-Decomposition-Based Analysis

Applicable for HRT DAG tasks

- Graham, Siam J. of Appl. Math., 1969
- Melani et al., ECRTS 2015
- He et al., RTSS 2022
- Fonseca et al., RTJ 2019





- Is there an SRT-optimal scheduler for a DAG task with self-dependencies that can be analyzed without decomposition?

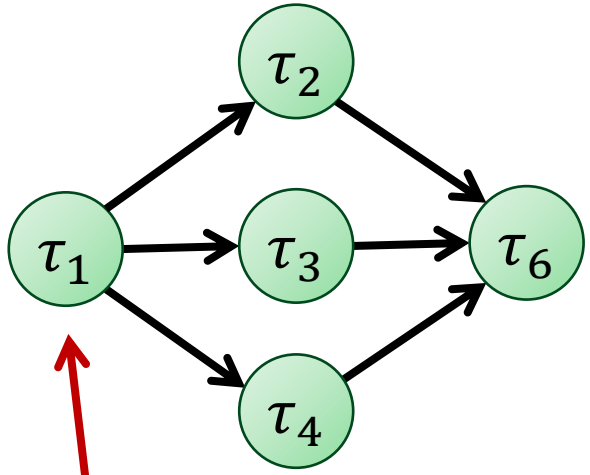
Challenge: Priority Assignment

How to set cross-instance job priorities?

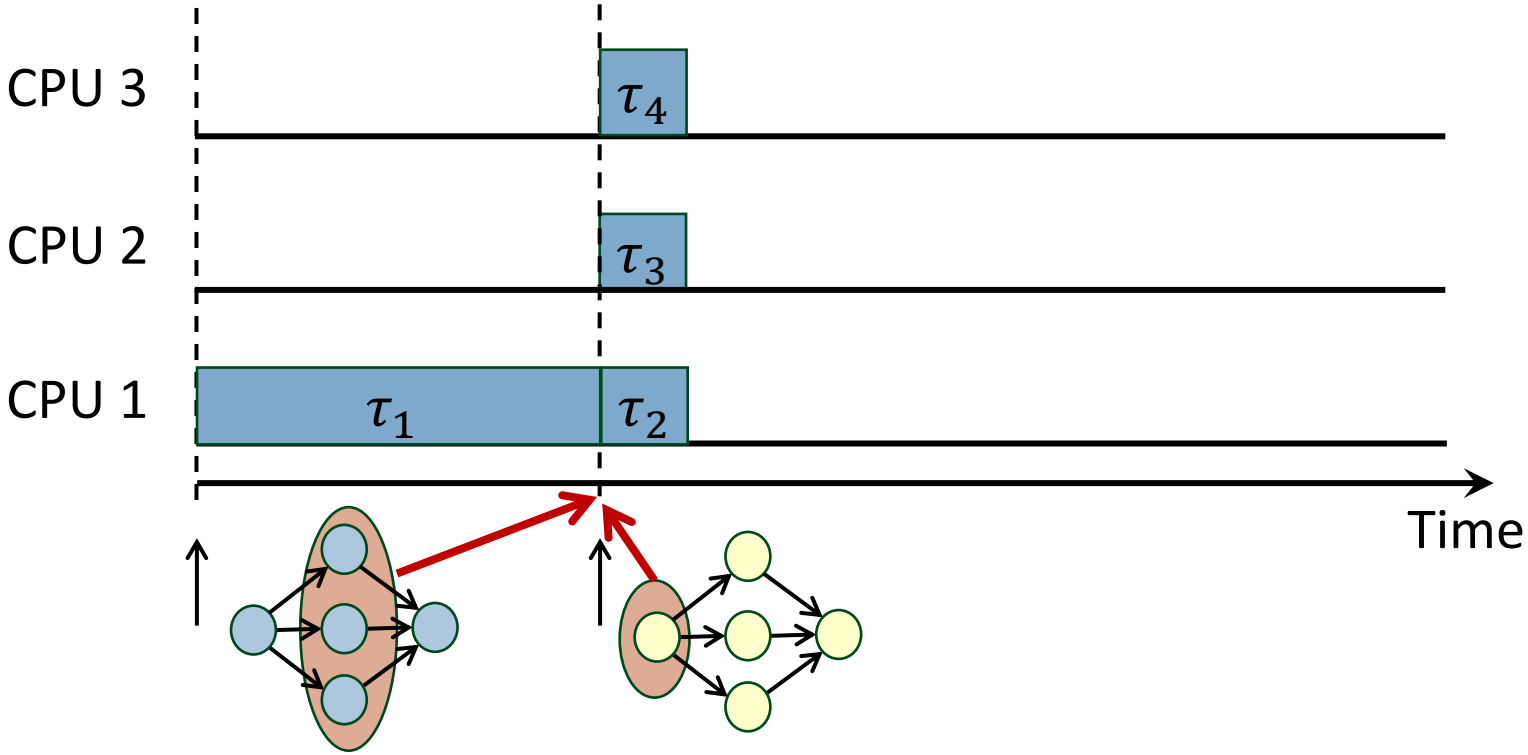
High priority → Low priority

τ_2 τ_3 τ_4 τ_1

GEDF/FIFO: prioritize jobs of earlier instances



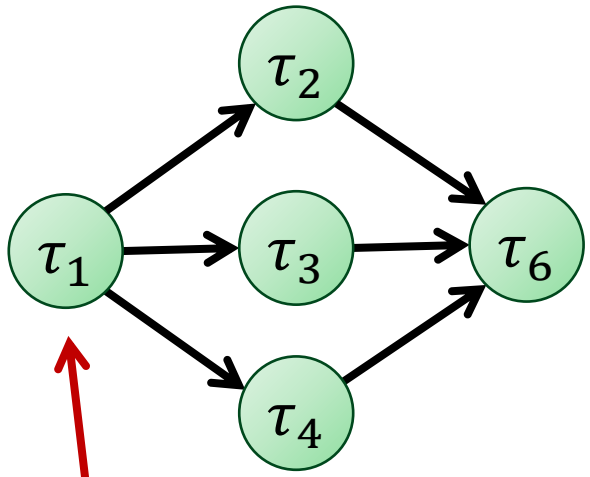
WCET = period



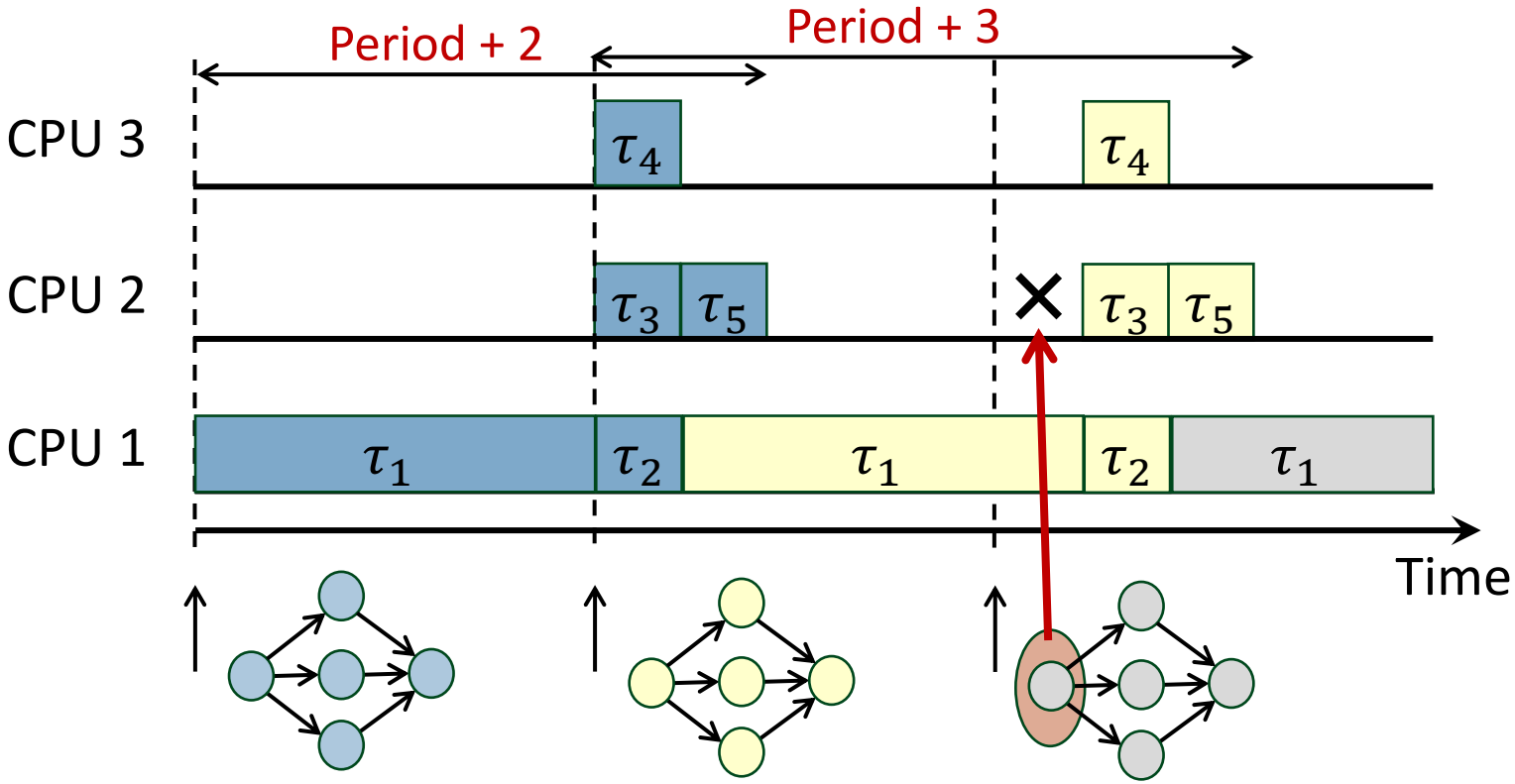
Challenge: Priority Assignment

How to set cross-instance job priorities?

GEDF/FIFO: prioritize jobs of earlier instances



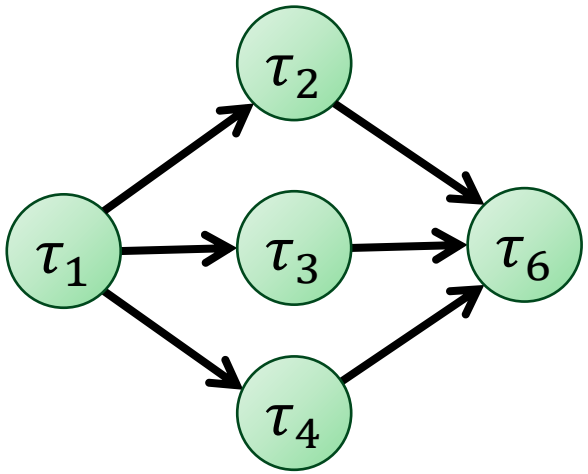
WCET = period

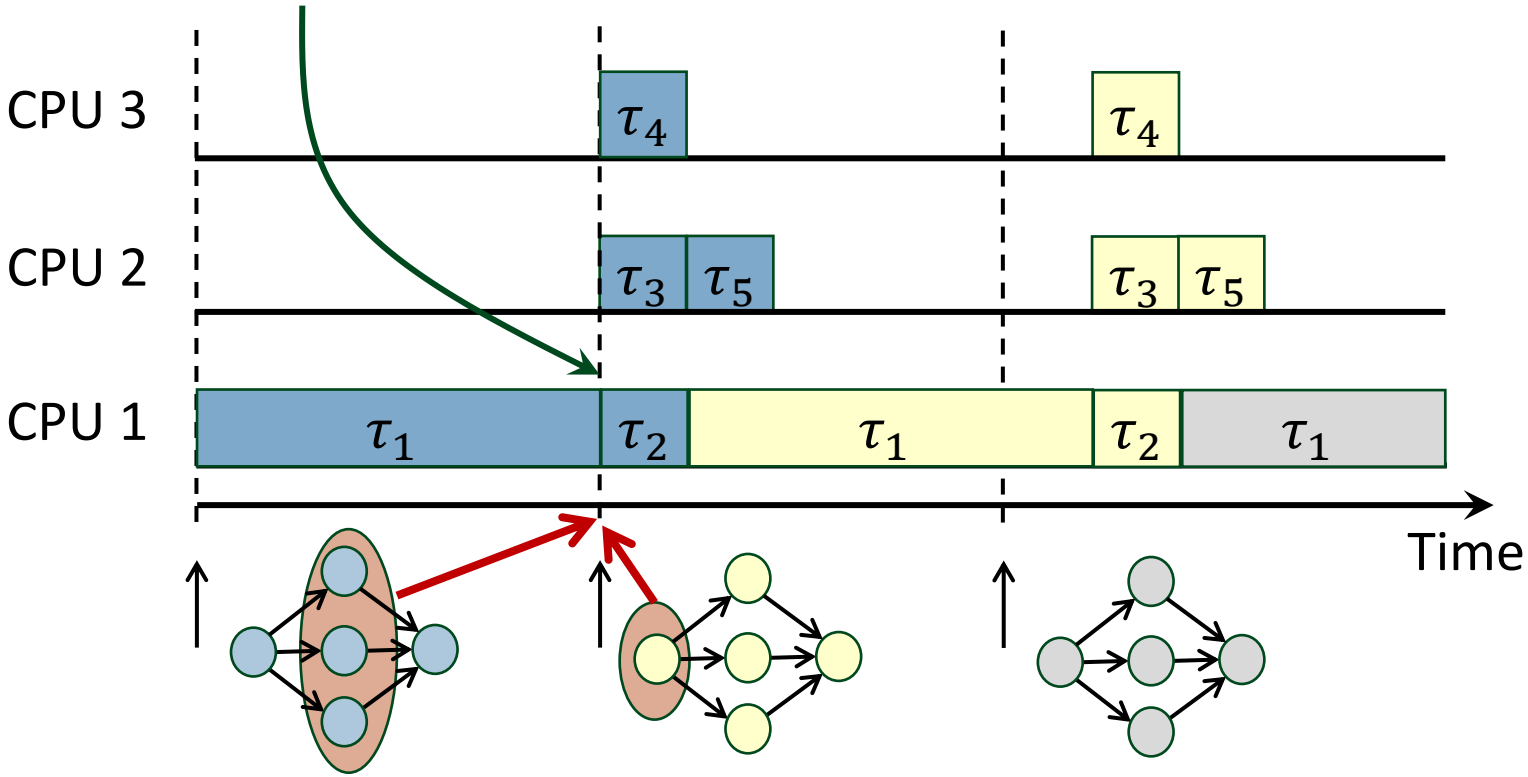


Priority Assignment

Goal: Ensure concurrent progress of multiple DAG instances

Give τ_1 high-enough priority to be scheduled

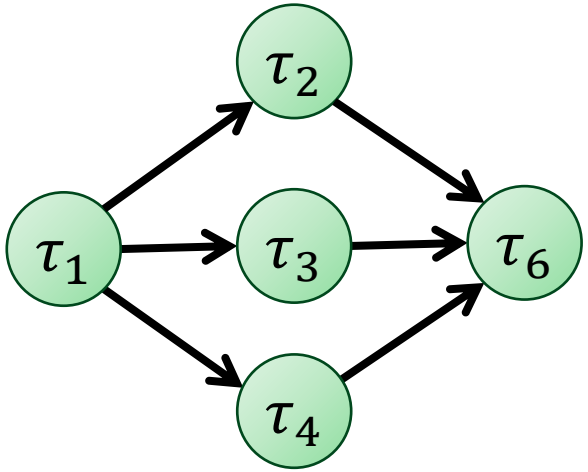


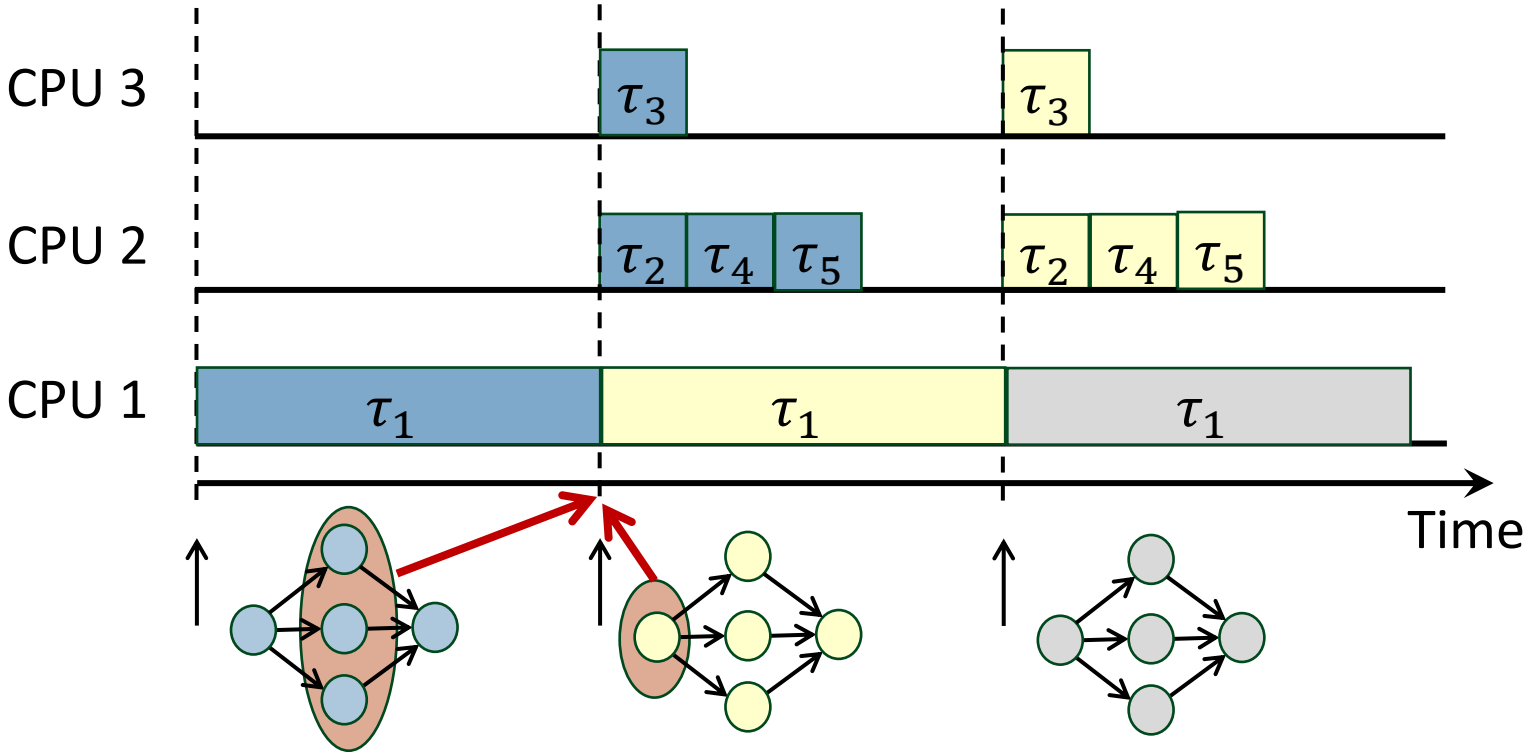


Priority Assignment

Goal: Ensure concurrent progress of multiple DAG instances

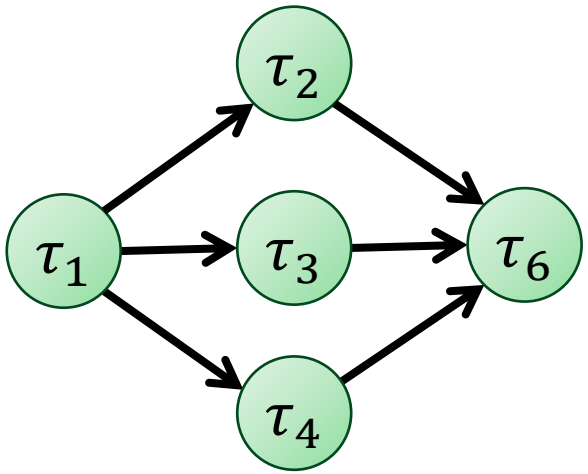
Give τ_1 high-enough priority to be scheduled

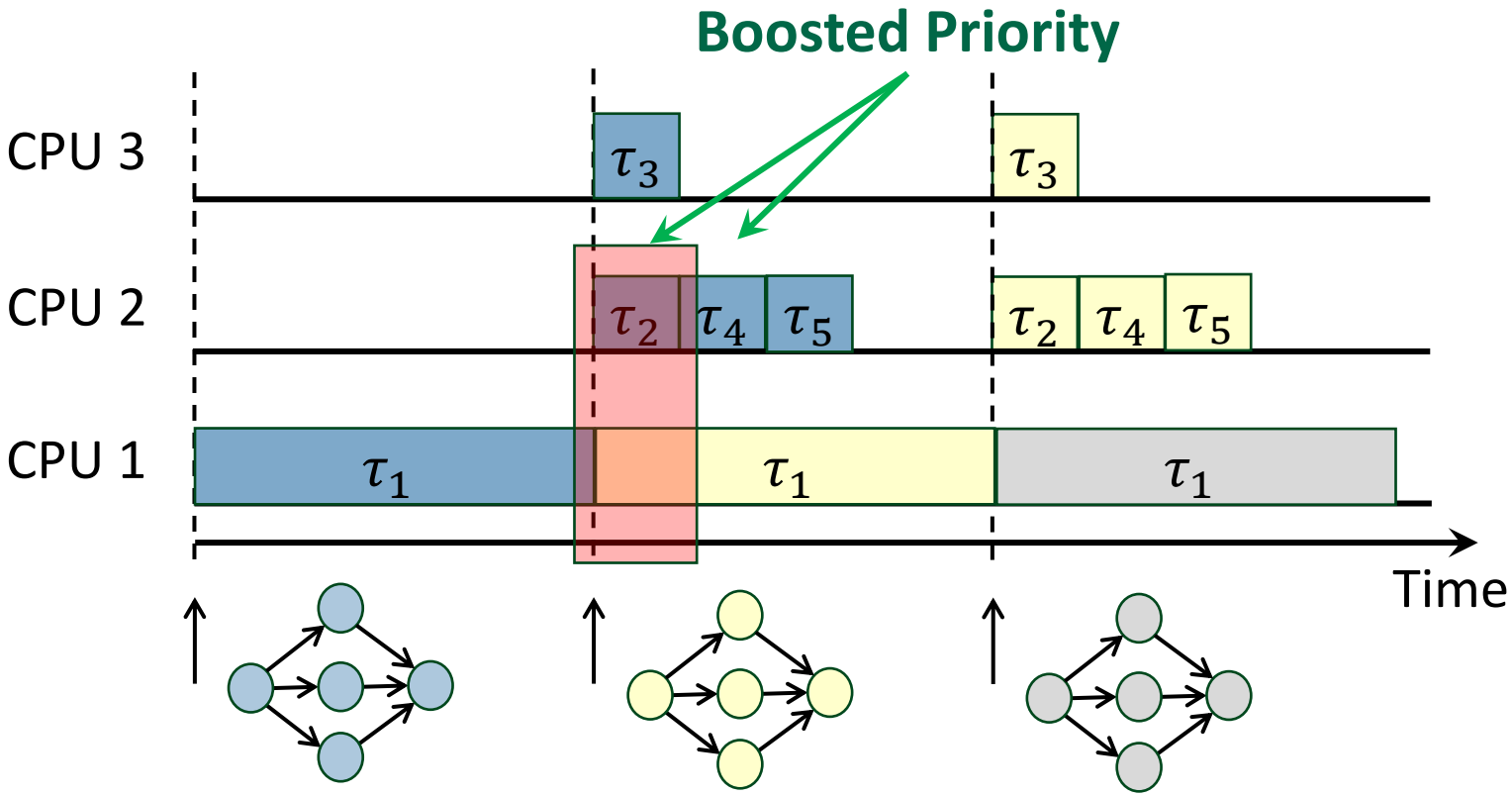




Priority Assignment

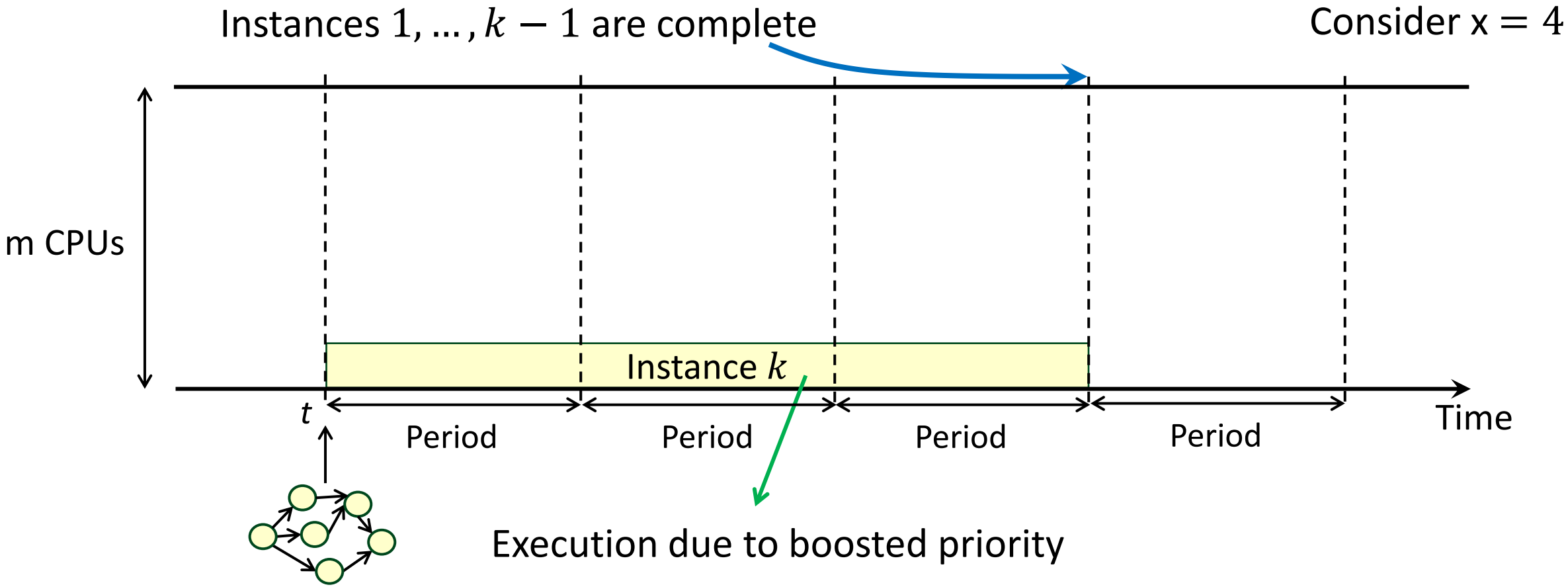
- Prioritize jobs of earlier instances
- But, the lowest-indexed pending job of each DAG instance is priority-boosted



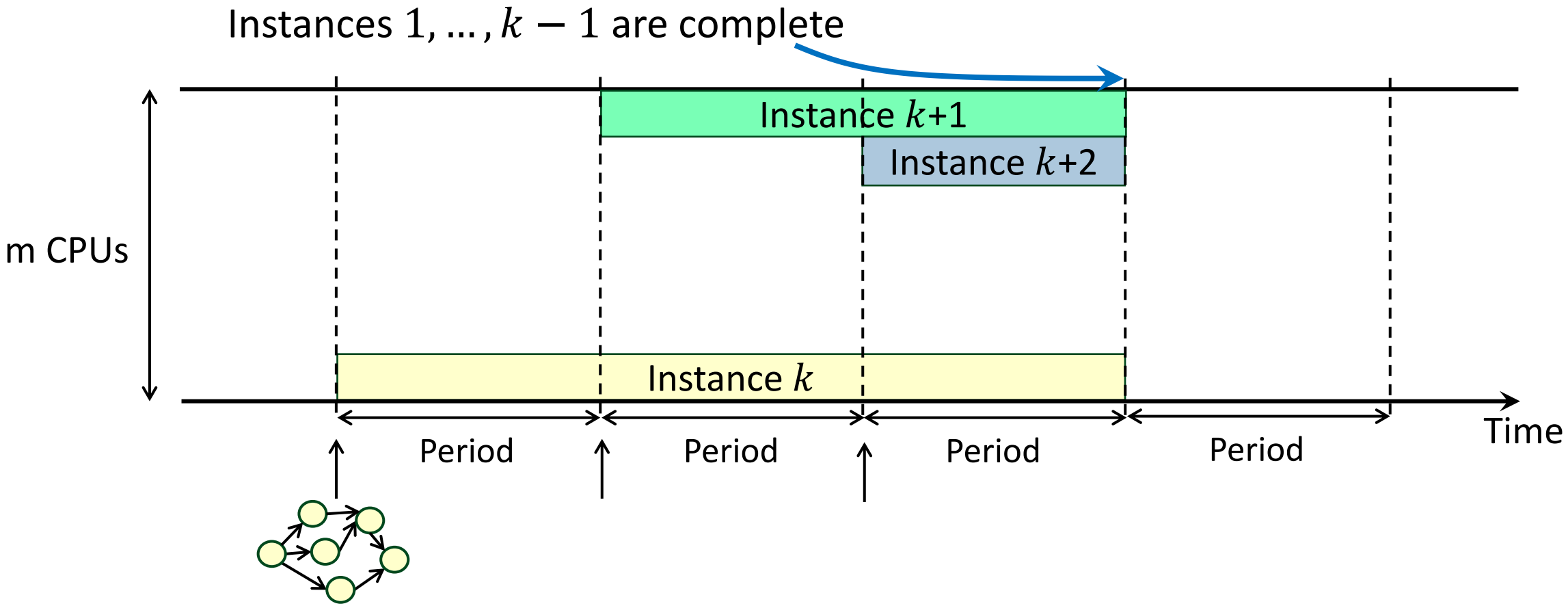


Response-Time Analysis

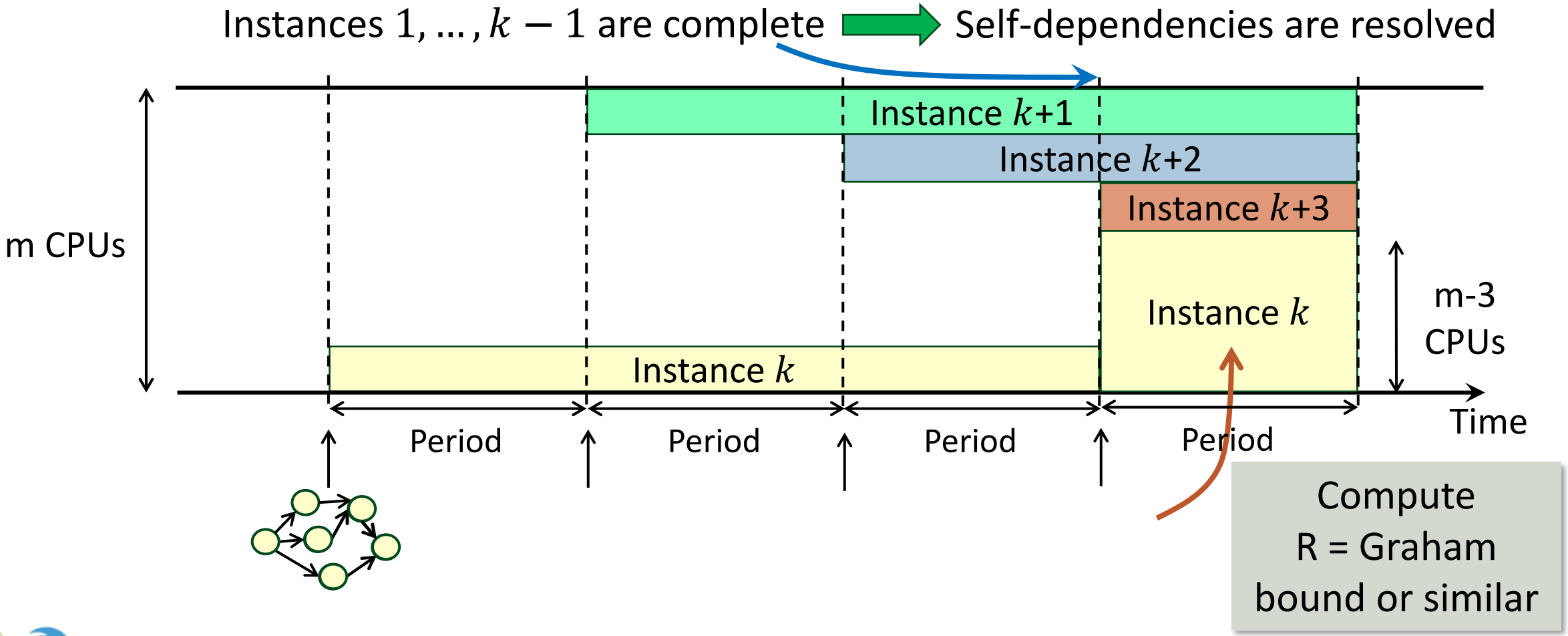
How can a DAG's response time be at most x·Period?



Response-Time Analysis

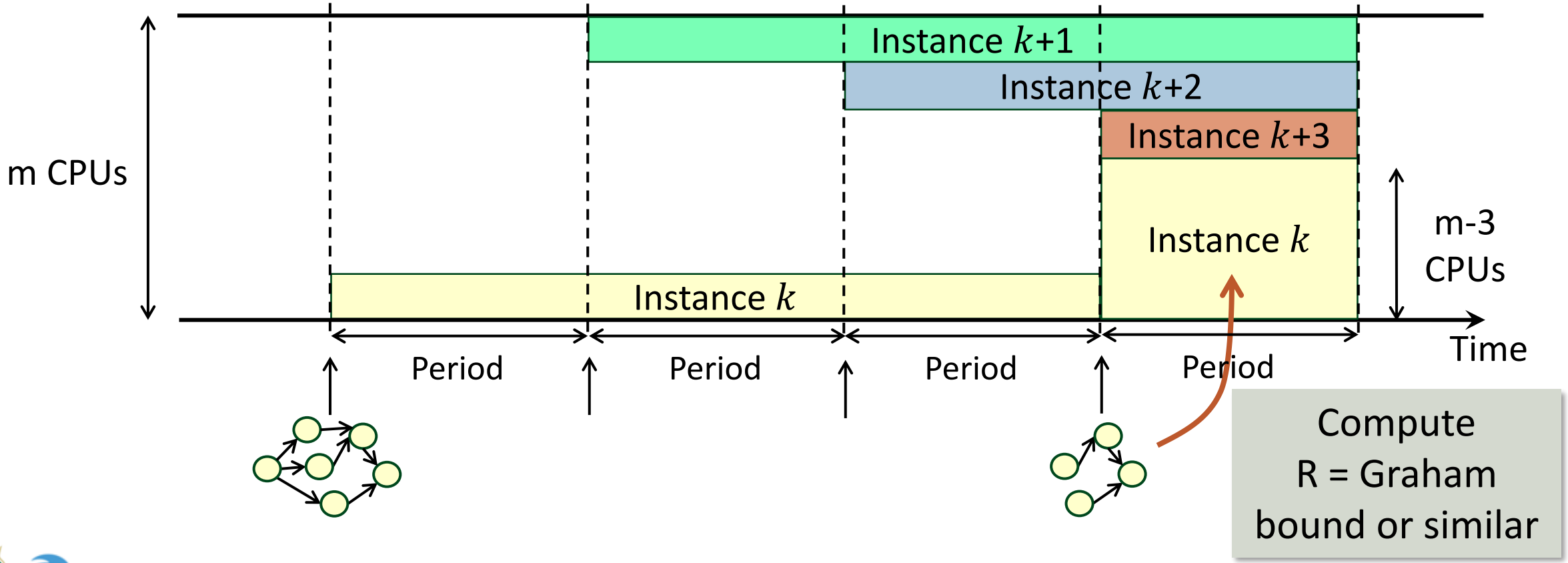


Response-Time Analysis



Response-Time Analysis

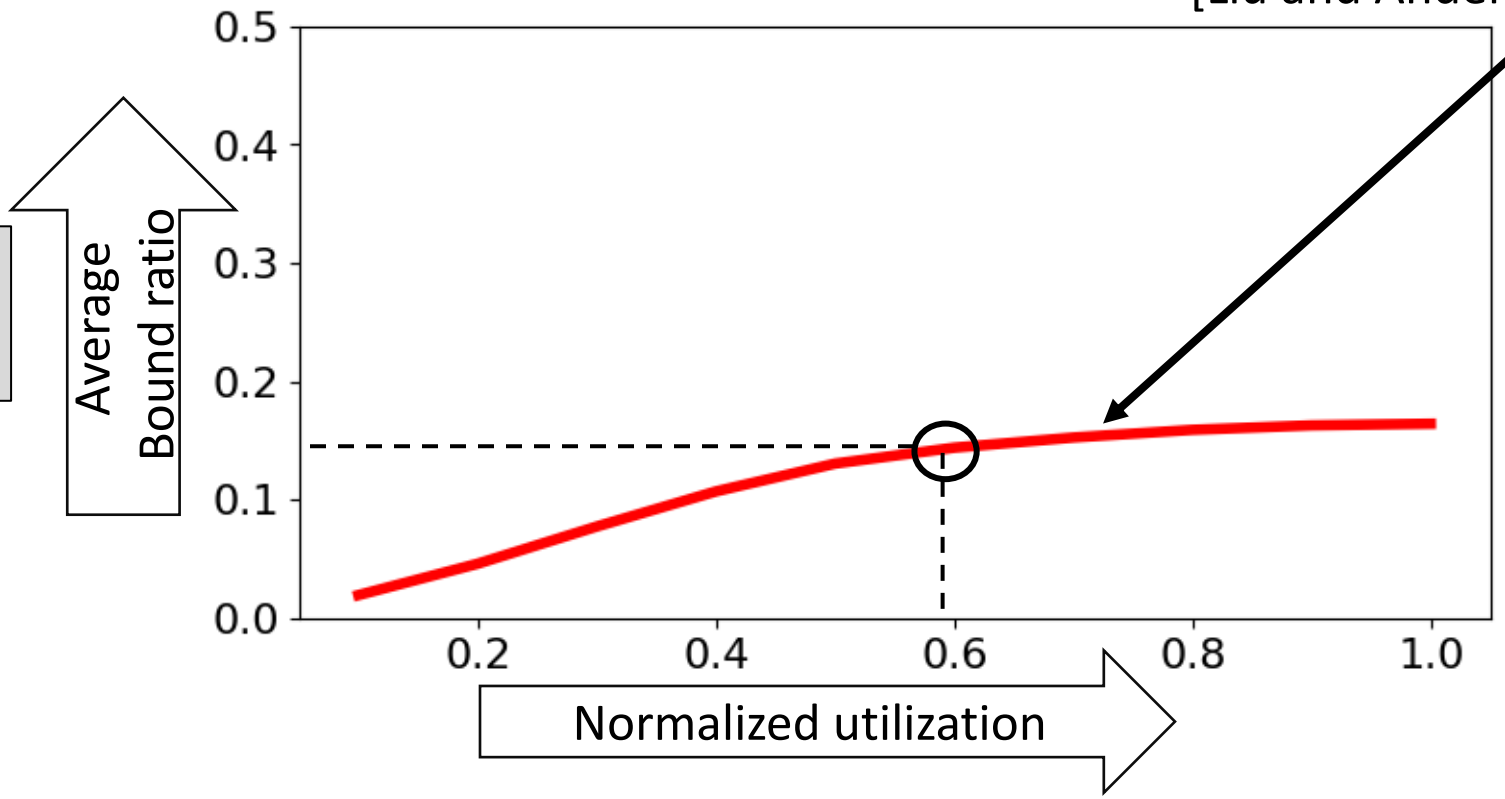
$R \leq \text{Period} \Rightarrow \text{Response time} \leq x \cdot \text{Period}$



Evaluation

Dependency level = 1

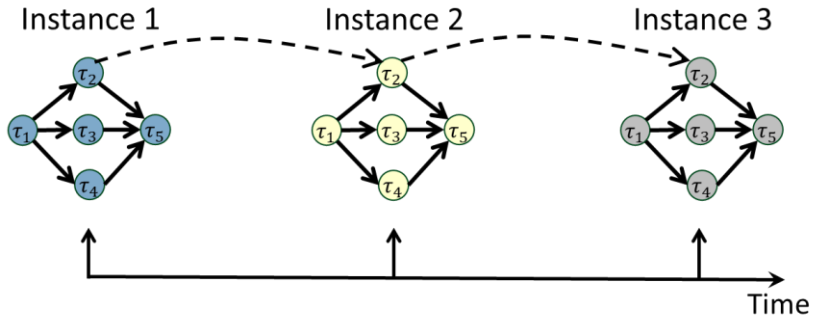
$$\frac{\text{Bound under our scheduler}}{\text{Bound under offset-based scheduler [Liu and Anderson, RTSS 2010]}}$$



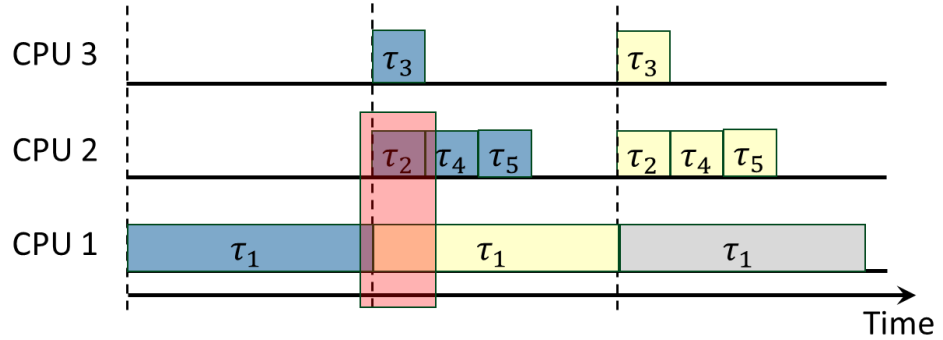
≤ 1 means lower bound

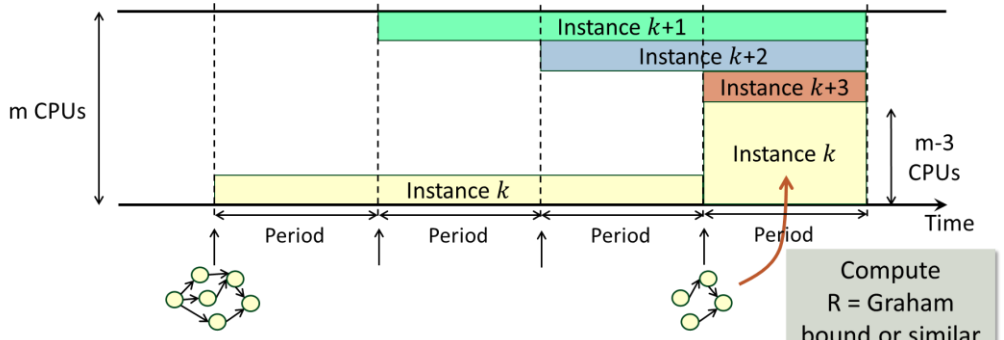
No dominance!

Conclusion & Thank You!



Dependency level = Parallelism level = 1





Future work

- Multi-DAG system
- Non-preemptive systems