A Problem of Time vs. Density Tradeoff in Multicore Fluid Scheduling

Kang-Wook Kim, Jeongyoon Eo, and Chang-Gun Lee
Seoul National University, Korea

2017. 6. 27

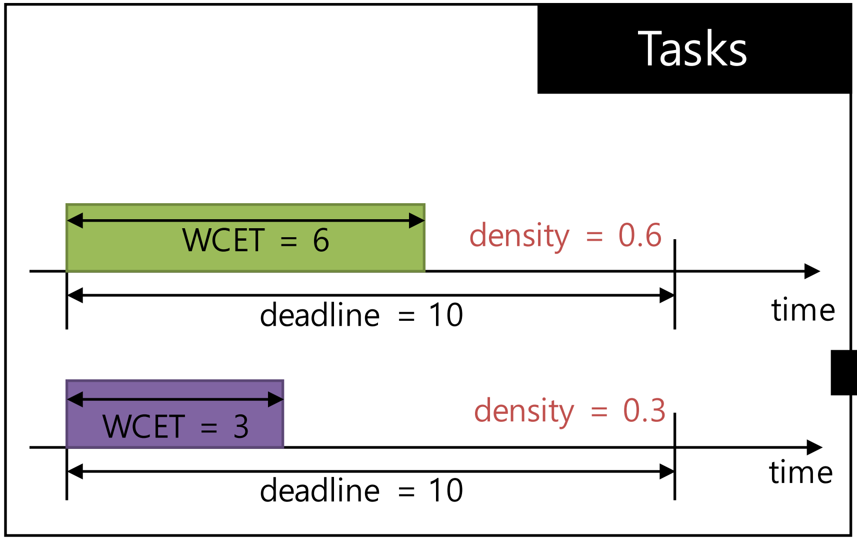
Fluid Scheduling

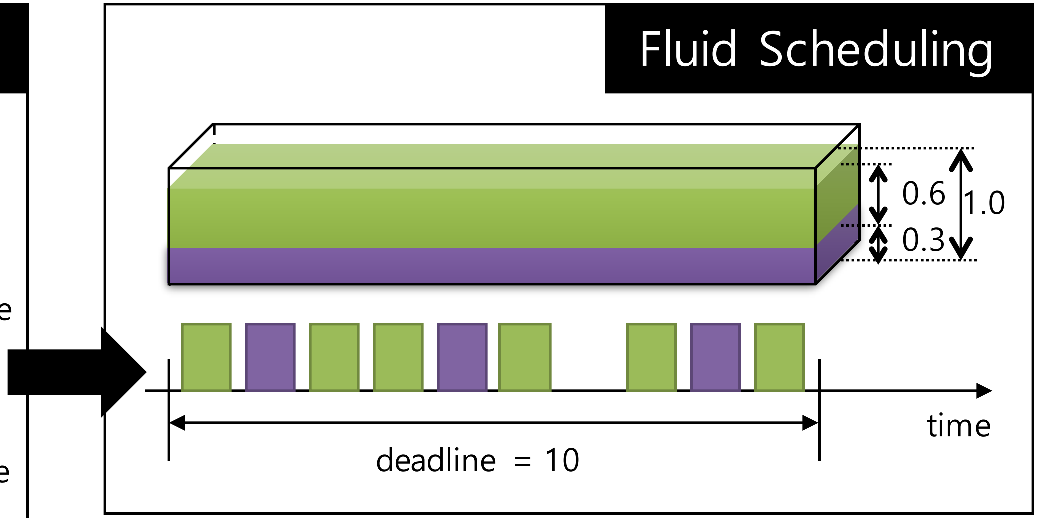
Fluid Scheduling

- Based on P-Fair (Proportionate Fairness)
- Schedule tasks proportional to their density

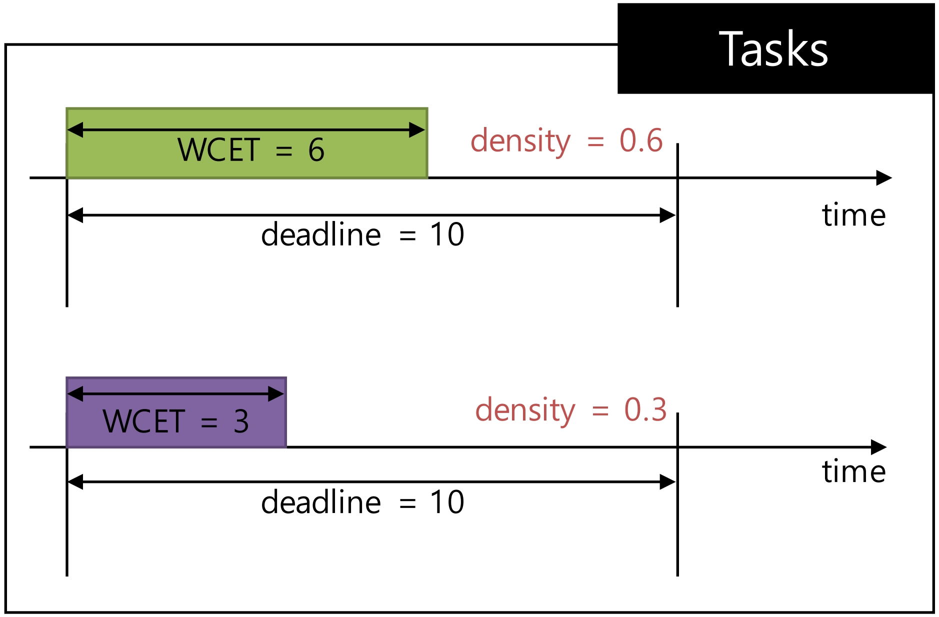
density = WCET / deadline

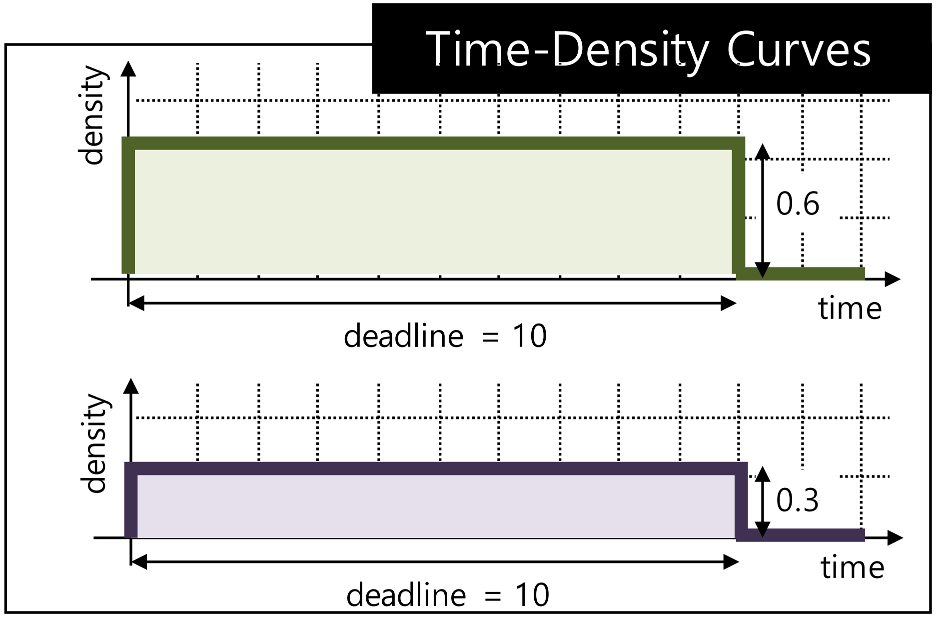
(WCET = Worst Case Execution Time)





Schedulability of Fluid Schedule

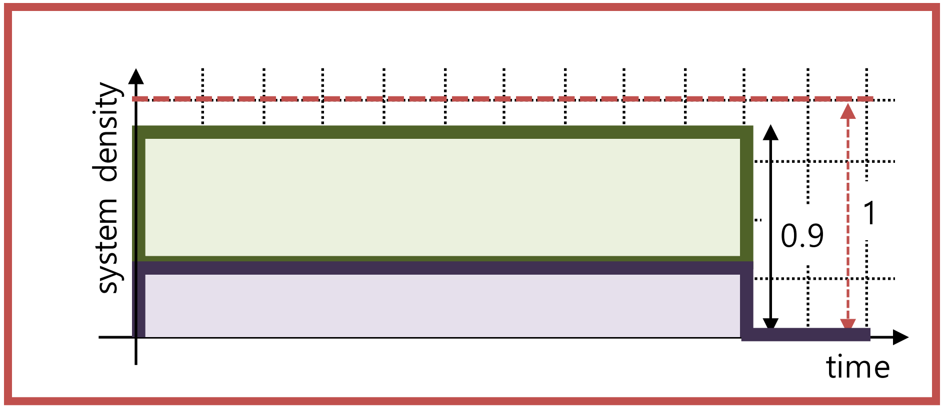




Schedulability of Fluid Schedule

- Tasks are **schedulable**,

  if **sum of densities** of all active tasks is **smaller or equal to the number of processors** at any time instant



Research Goal

Maximize schedulability of
real-time multicore fluid scheduling

Minimize system peak density of
fluid scheduling

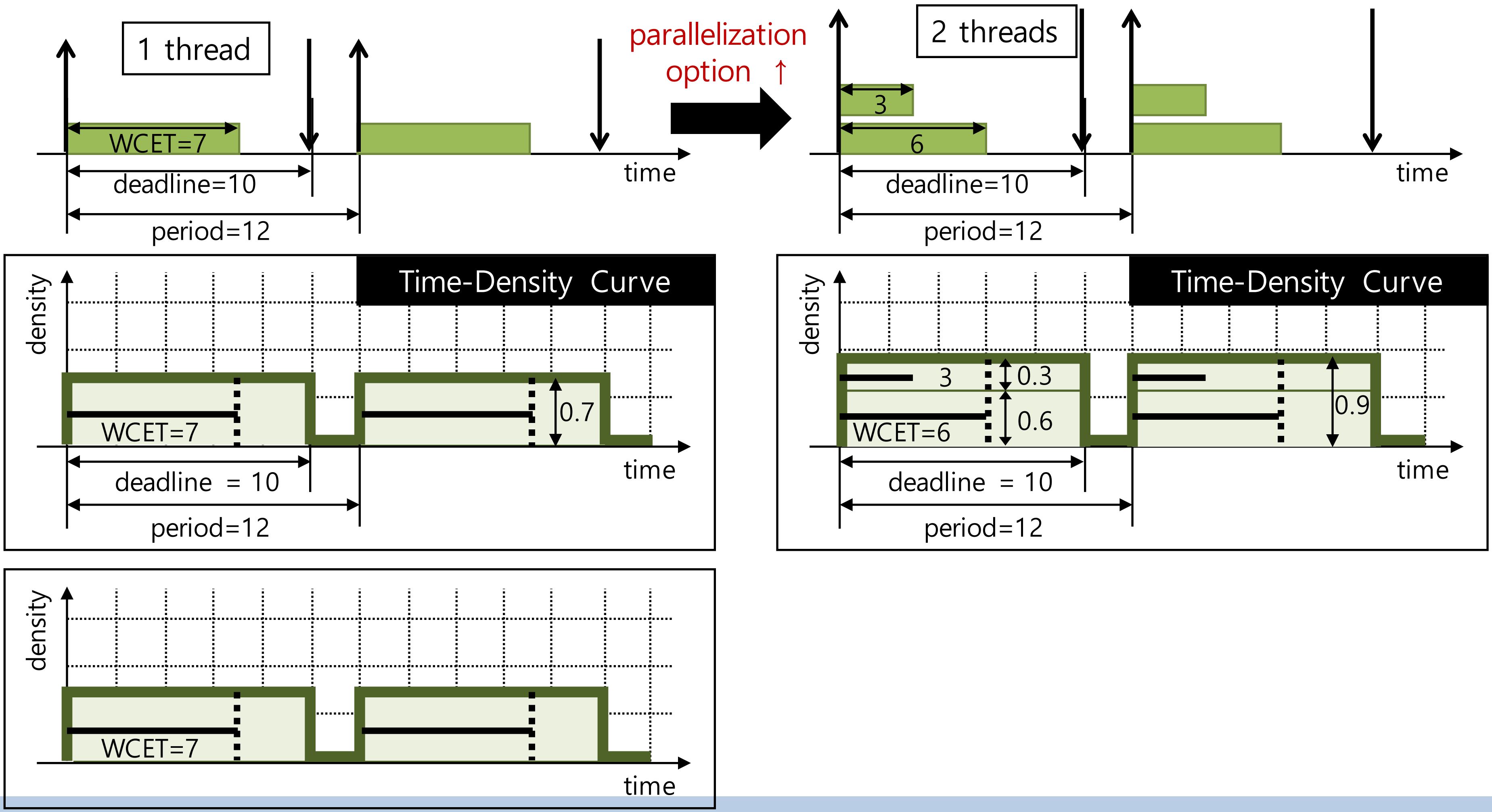
by controlling

1) **parallelization option** (number of threads)
2) **artificial deadline** (shorter than original deadline)
3) **artificial period** (shorter than original period)
4) **task offset**

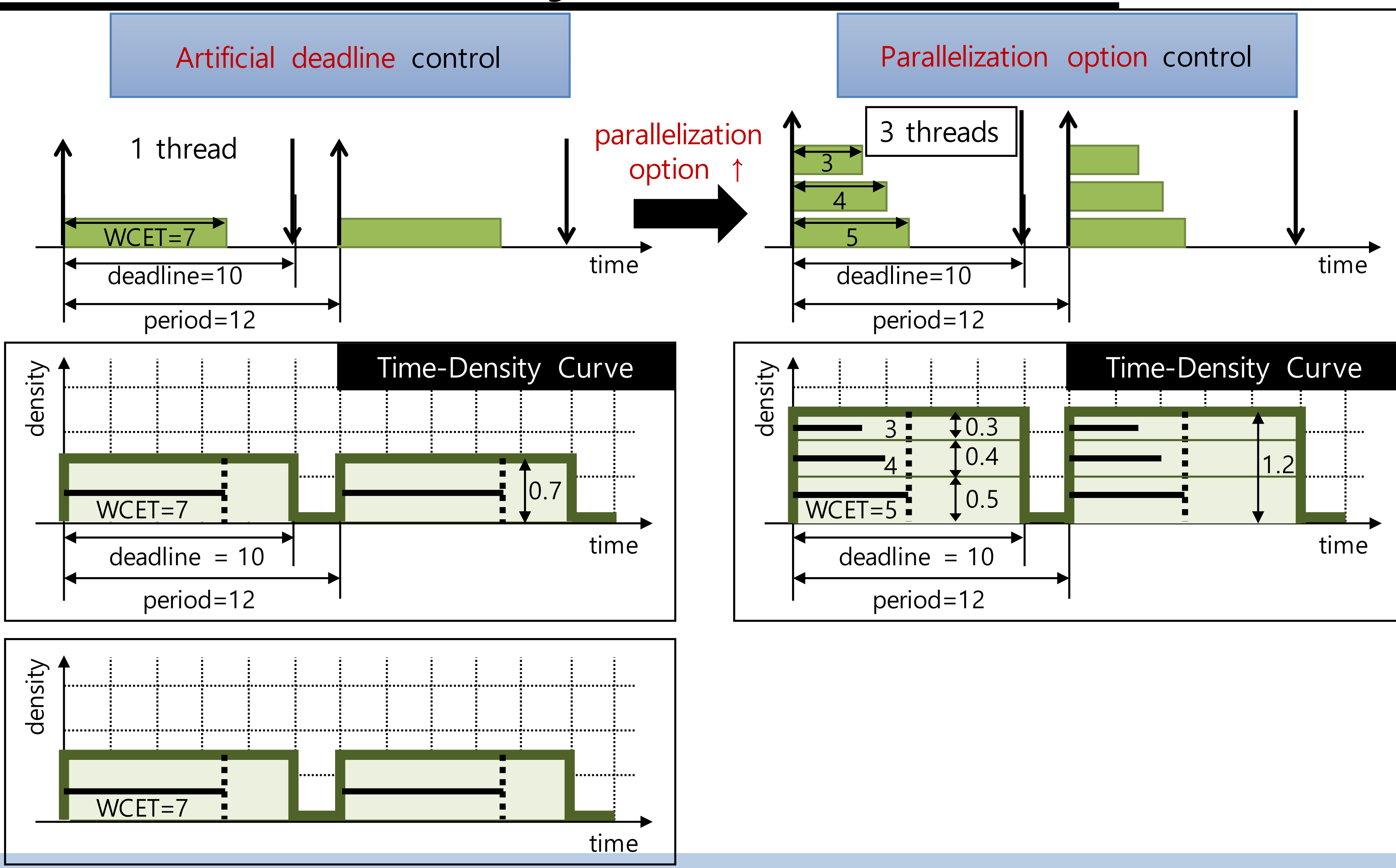
Open Problem 1: Time vs. Density Tradeoff

Artificial deadline control

Parallelization option control

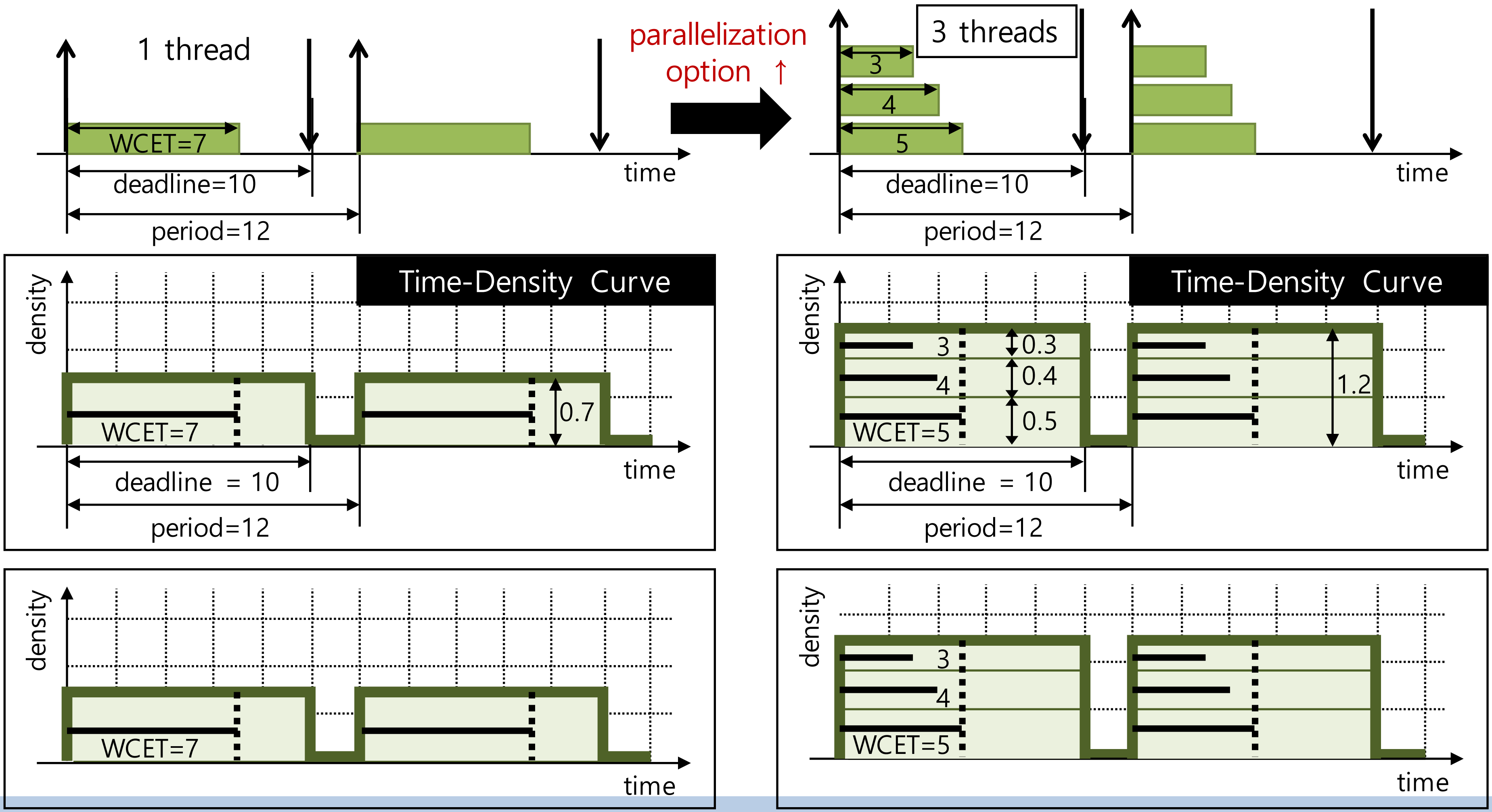


Open Problem 1: Time vs. Density Tradeoff



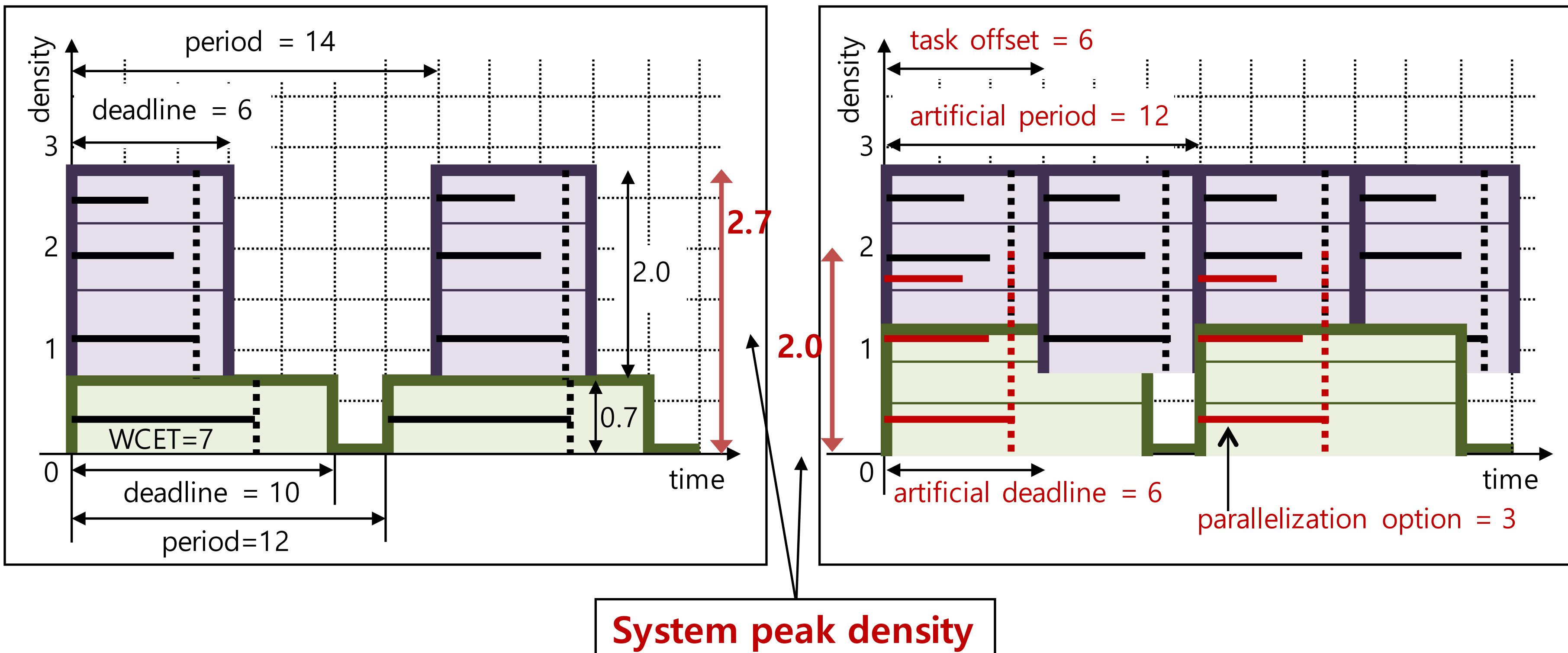
Open Problem 1: Time vs. Density Tradeoff

Artificial deadline and parallelization option control



Open Problem 2: System-wide Tradeoff?

artificial deadline and parallelization option
+
artificial period and task offset control

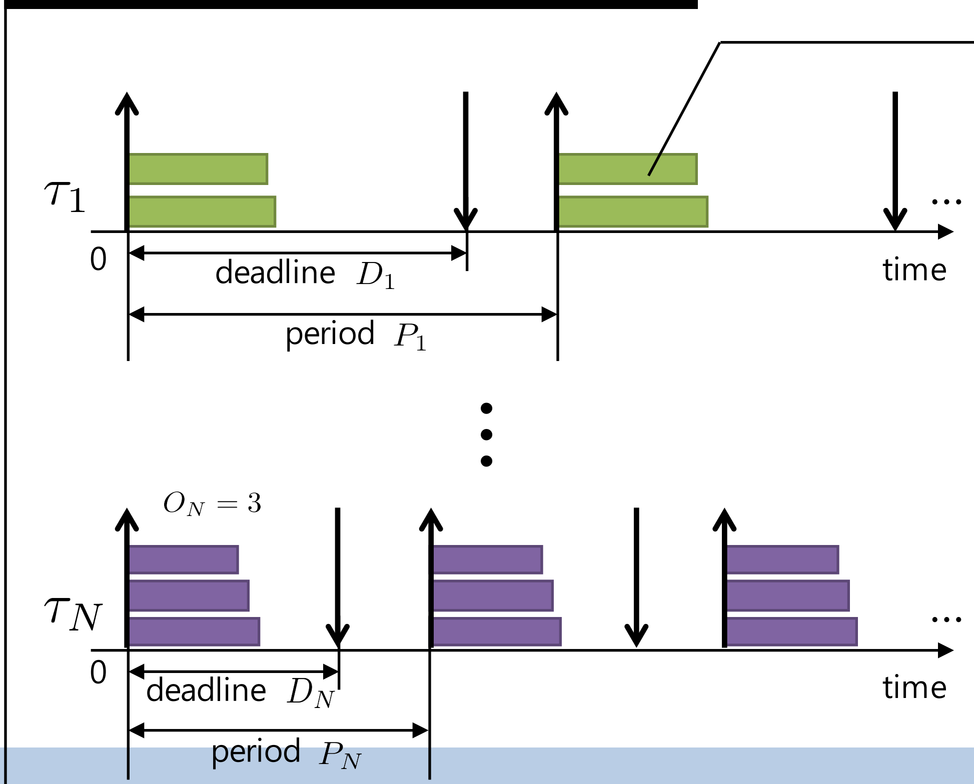


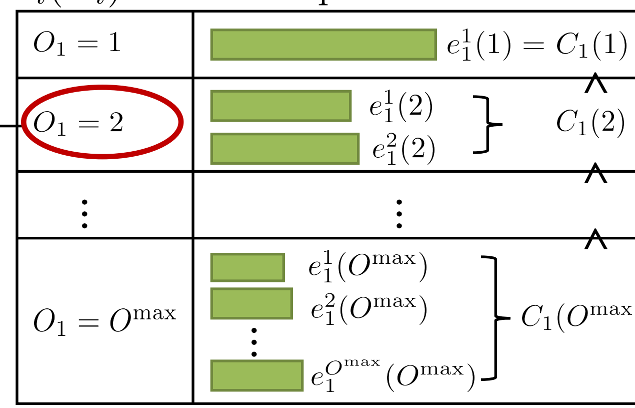
Problem Formulation (1 / 2)

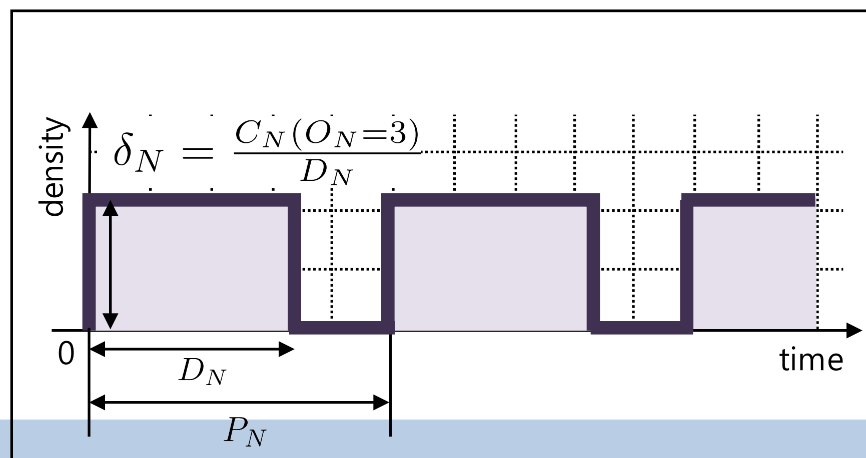
System with M homogeneous CPU cores,
N parallelizable periodic tasks
$\tau_i = (P_i, D_i, C_i(O_i))$

P_i: original period
D_i: original deadline
O_i: parallelization option
$e_i^k(O_i)$: WCET of k-th thread when parallelized into O_i threads
$C_i(O_i)$: total computation amout

Parallelizable Periodic Tasks

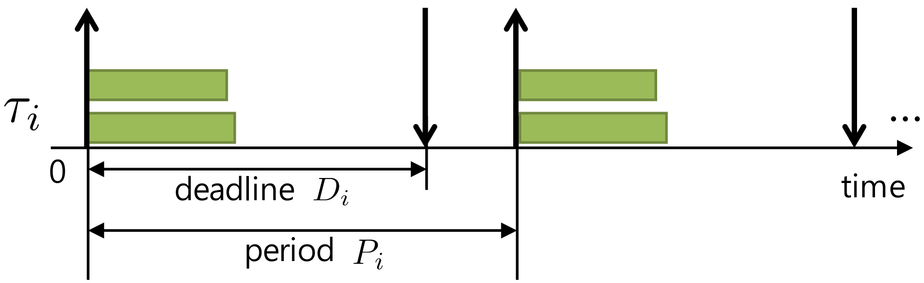


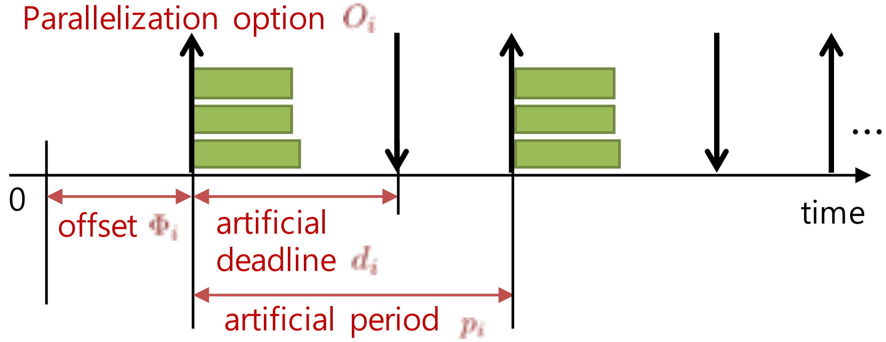




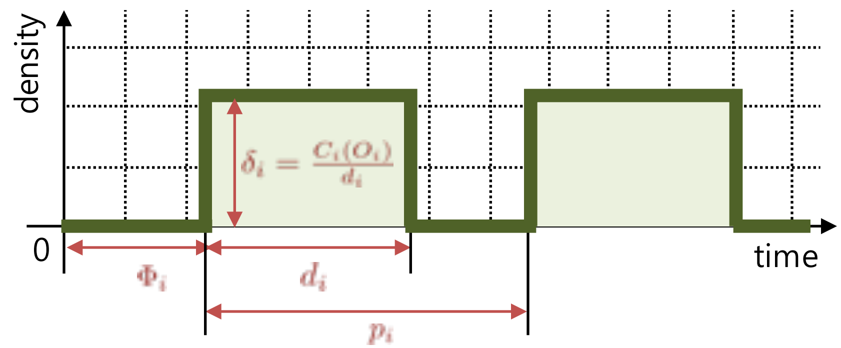
Problem Formulation (2 / 2)

Four control parameters





d_i: artificial deadline ($d_i < D_i$)
O_i: parallelization option
p_i: artificial period ($p_i < P_i$)
Φ_i: task offset

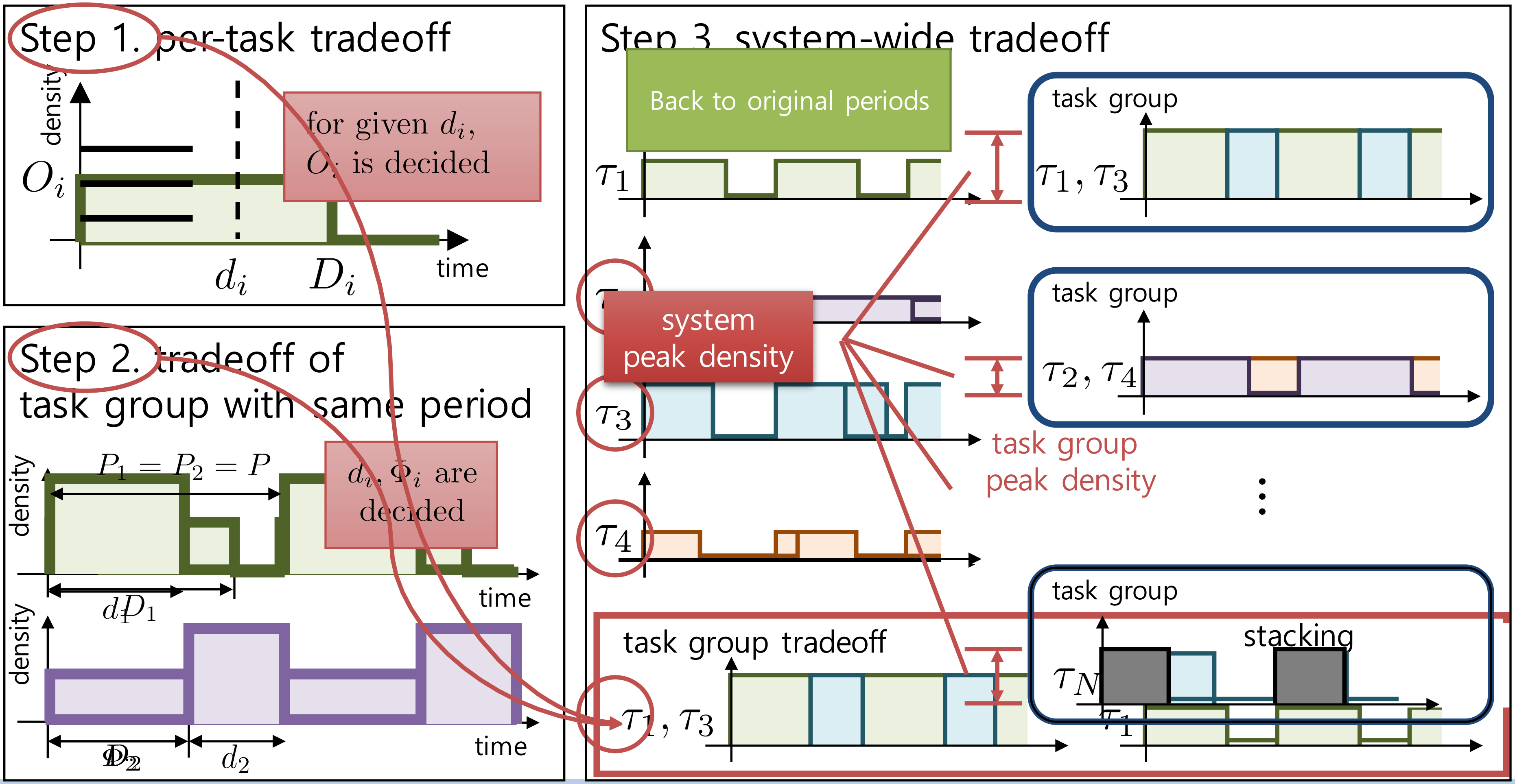


$$\delta_i((\Phi_i, O_i, d_i, p_i), t) = \begin{cases} C_i(O_i)/d_i & (t - \Phi_i) \bmod p_i \leq d_i \\ 0 & (t - \Phi_i) \bmod p_i > d_i \end{cases}$$

$$\text{minimize} \quad \max_{0 \leq t < HP} \sum_{i=1}^{N} \delta_i((\Phi_i, O_i, d_i, p_i), t)$$

Preliminary Solution

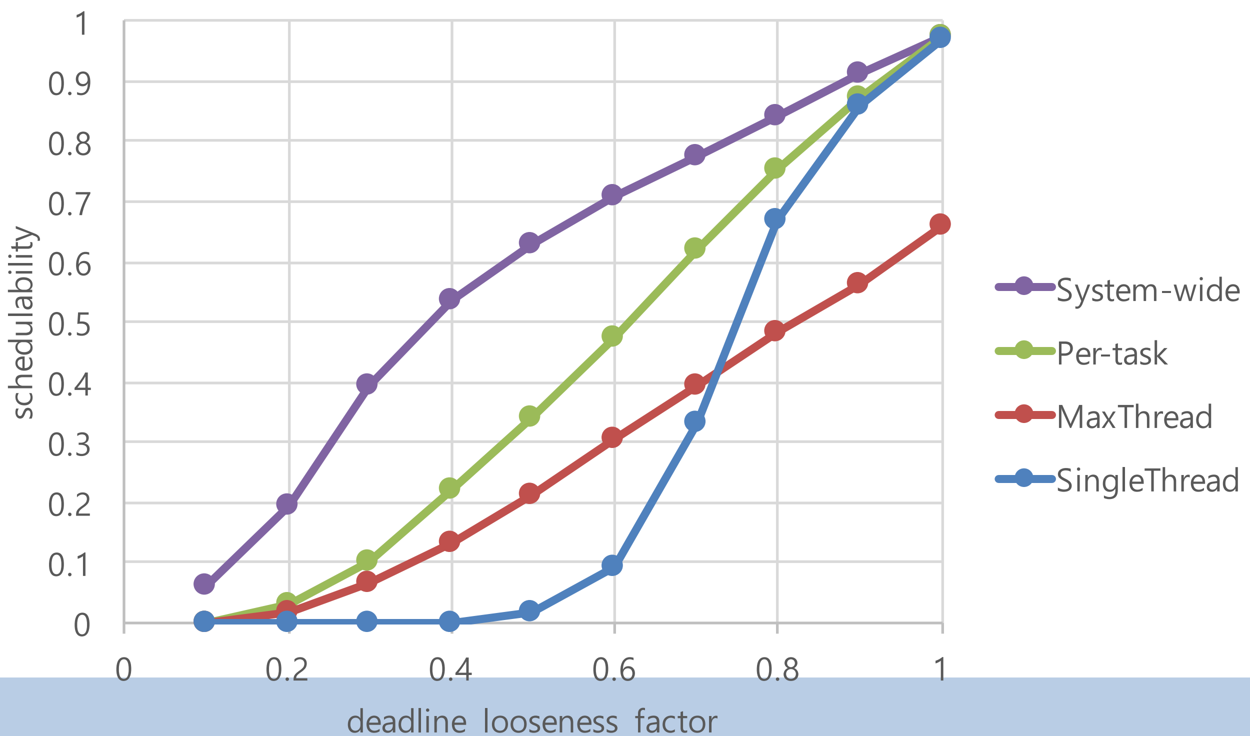
3-step approach



Simulation Result

Simulation Environment

1000 task sets
N = [3, 15]
M = 8
deadline looseness factor = deadline / period



Remaining Issues

- Considering Inter-core Memory Interference

- Complicated Task Model
  - Multi-segment task model
  - Fork-join task model

- Tradeoff in G-EDF

https://github.com/rubis-lab/RealTimeTaskSimulator

Thank you