Energy-aware Precise Scheduling of Mixed-Criticality Tasks

Ashikahmed Bhuiyan

University of Central Florida

Mixed-Criticality System

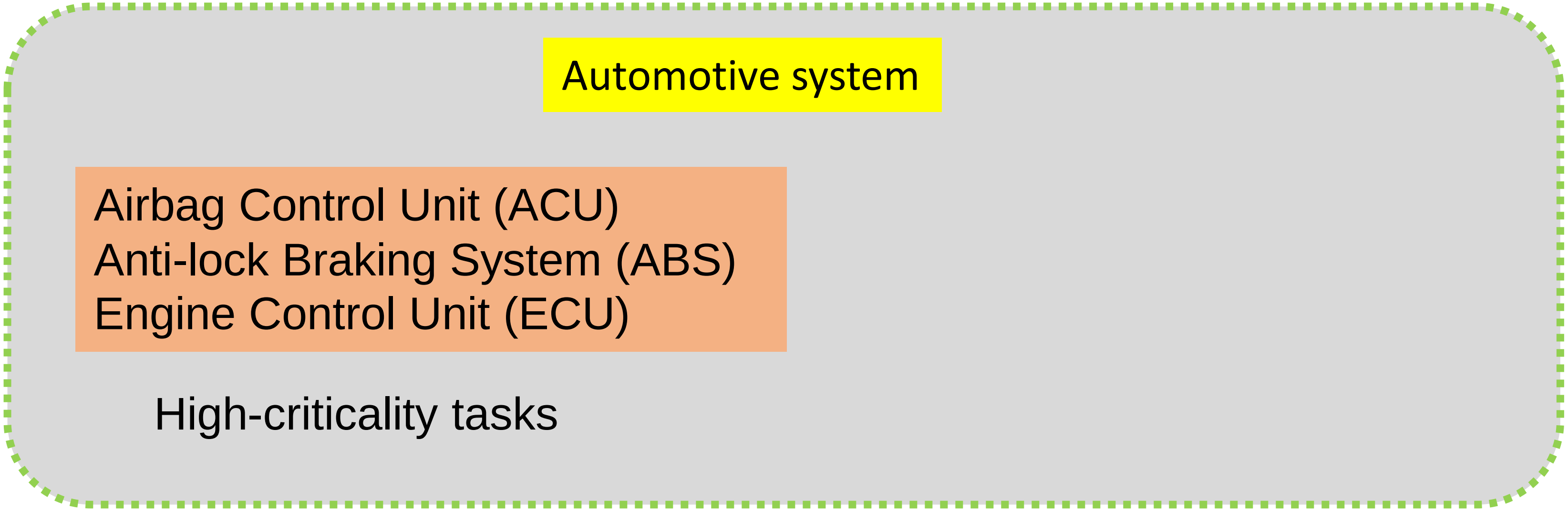
- Many of the modern embedded systems execute tasks with different criticalities.

Mixed-Criticality System

- Many of the modern embedded systems execute tasks with different criticalities.
- Mixed-Criticality (MC) model offers the feature to integrate system components with different assurance levels.

Mixed-Criticality System

- Many of the modern embedded systems execute tasks with different criticalities.
- Mixed-Criticality (MC) model offers the feature to integrate system components with different assurance levels.



Mixed-Criticality System

- Many of the modern embedded systems execute tasks with different criticalities.
- Mixed-Criticality (MC) model offers the feature to integrate system components with different assurance levels.

Automotive system

Airbag Control Unit (ACU)
Anti-lock Braking System (ABS)
Engine Control Unit (ECU)

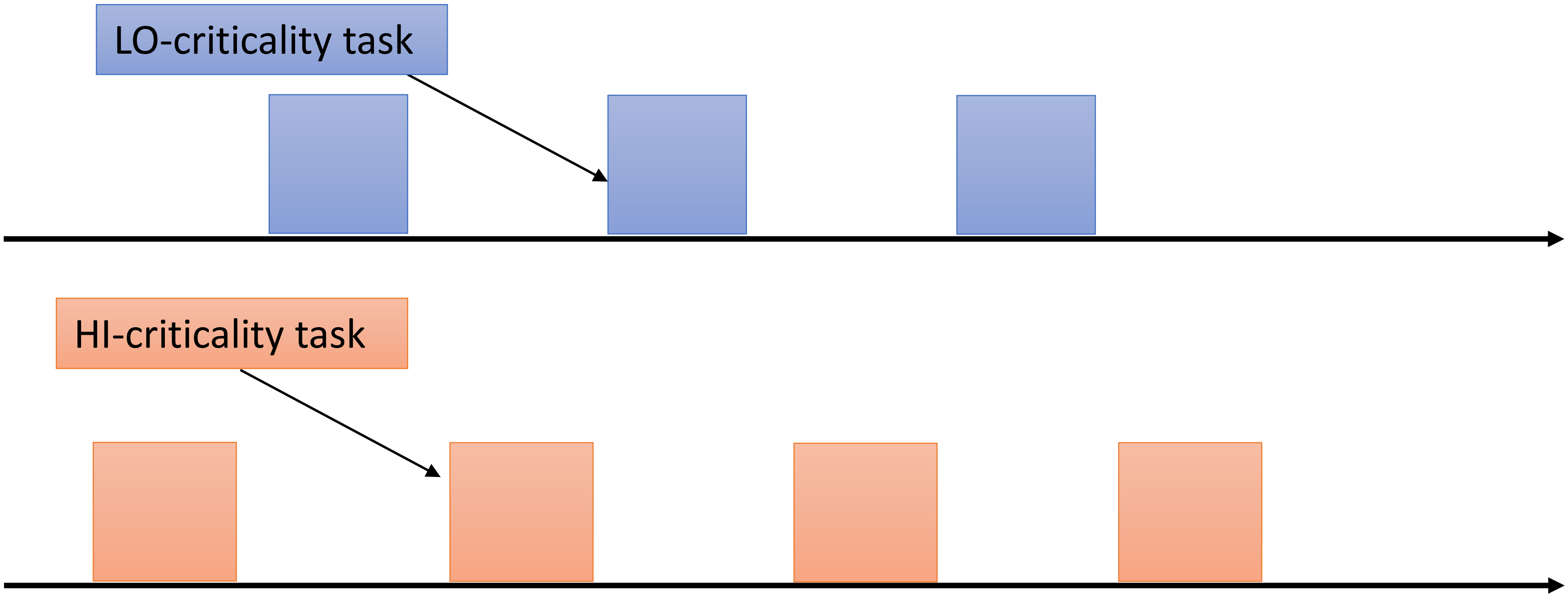
Air Conditioning Unit
Car infotainment system

High-criticality tasks

Low-criticality tasks

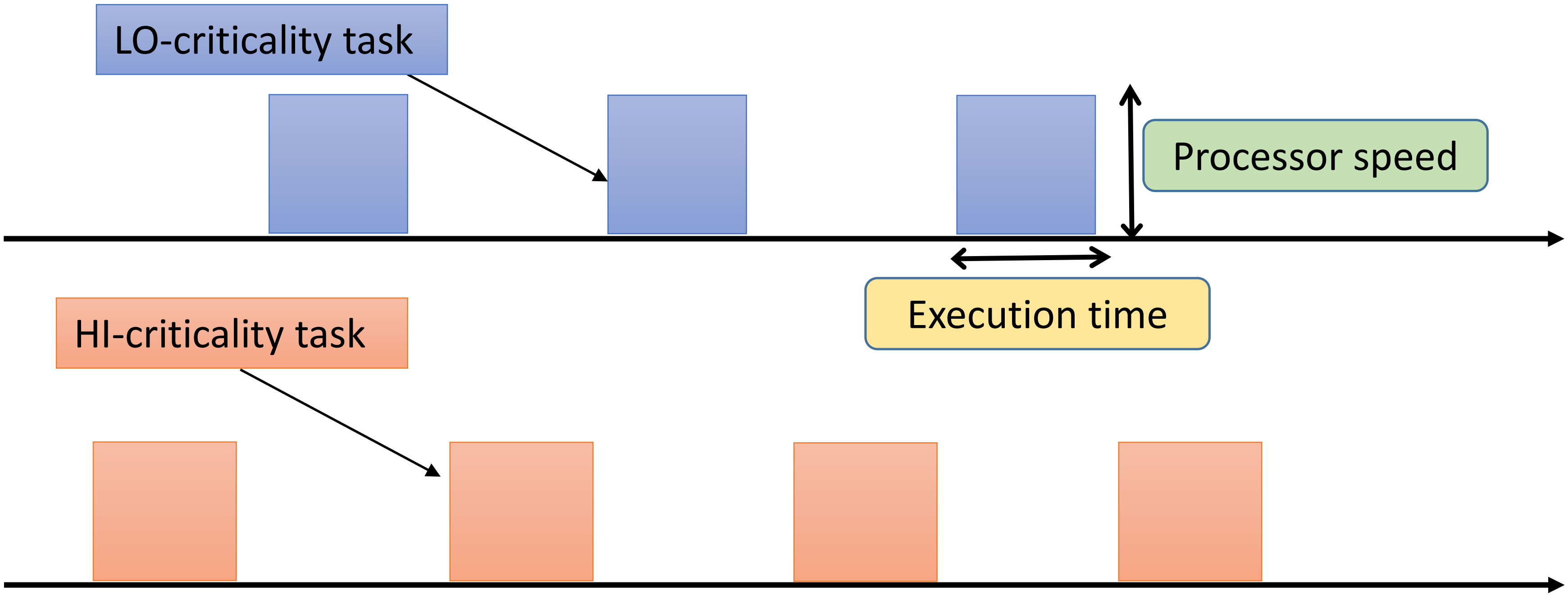
Problem and Motivation

- Traditional MC task model



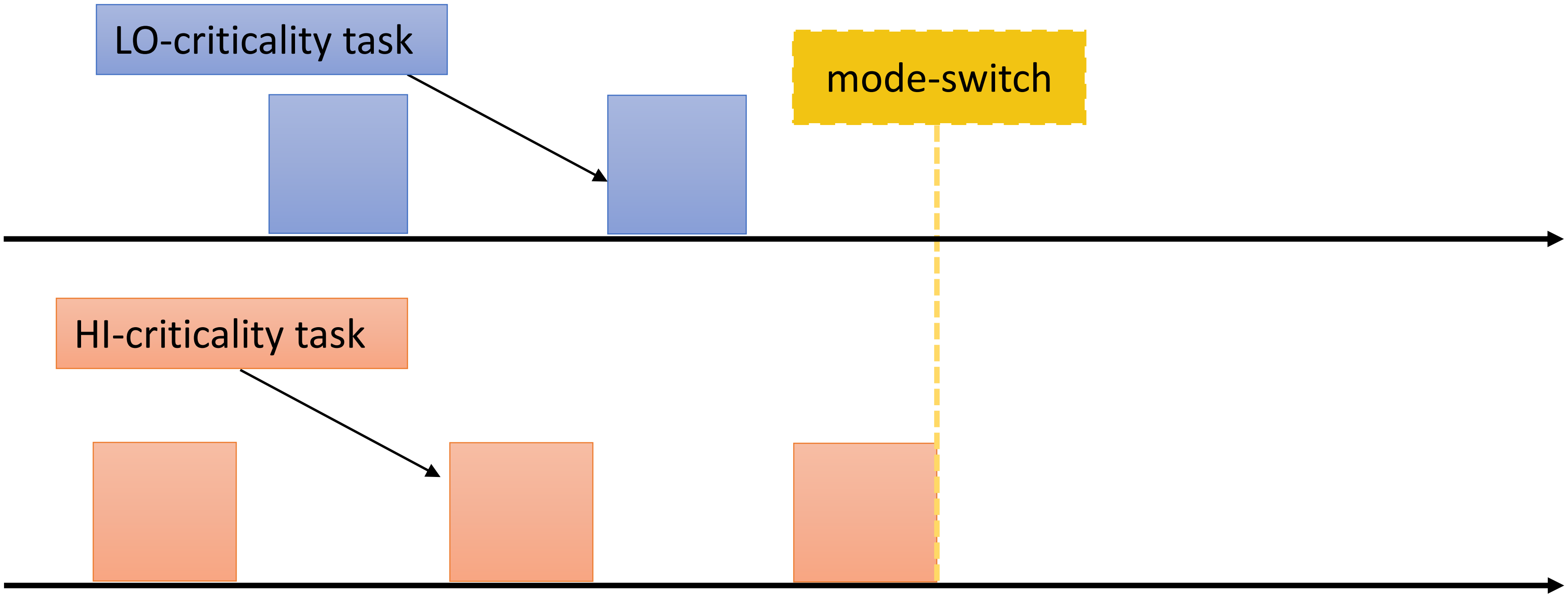
Problem and Motivation

- Traditional MC task model



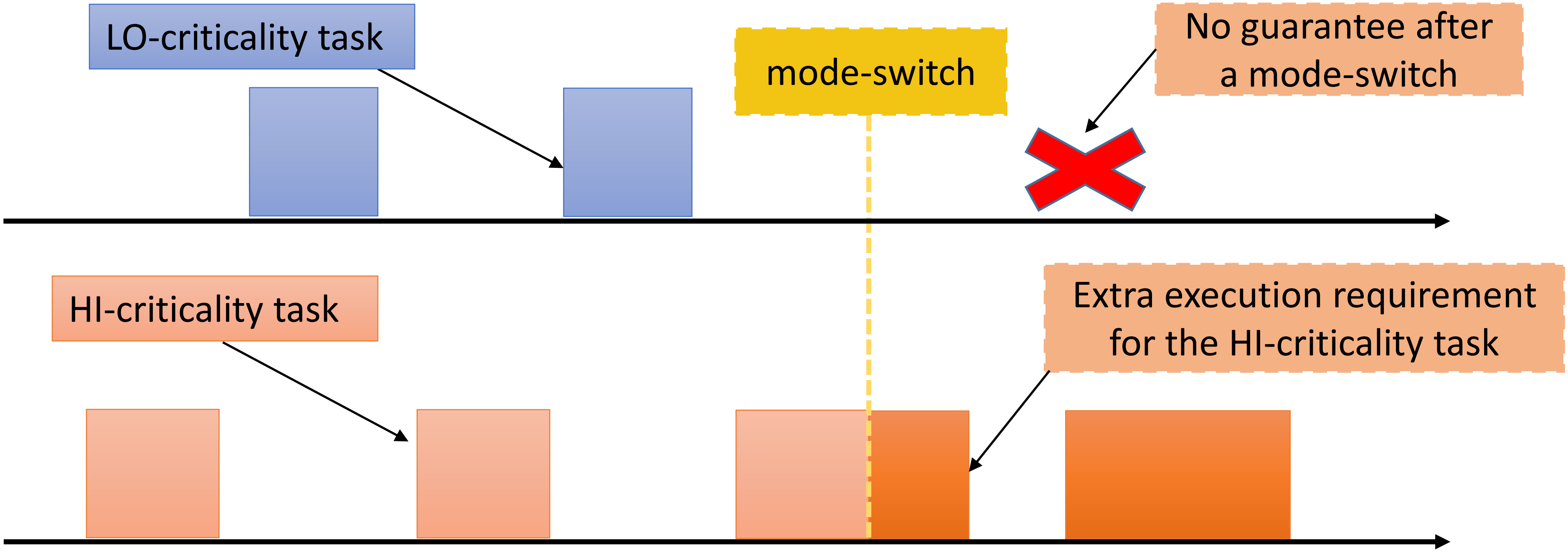
Problem and Motivation

- Traditional MC task model



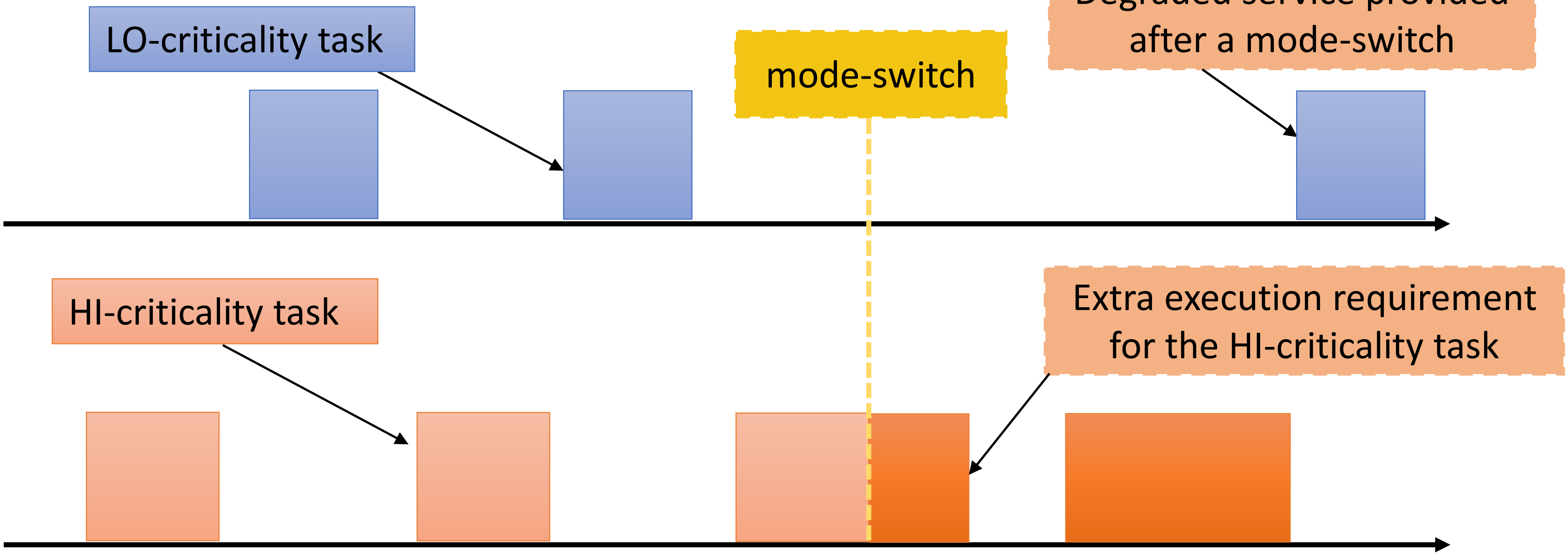
Problem and Motivation

- Traditional MC task model



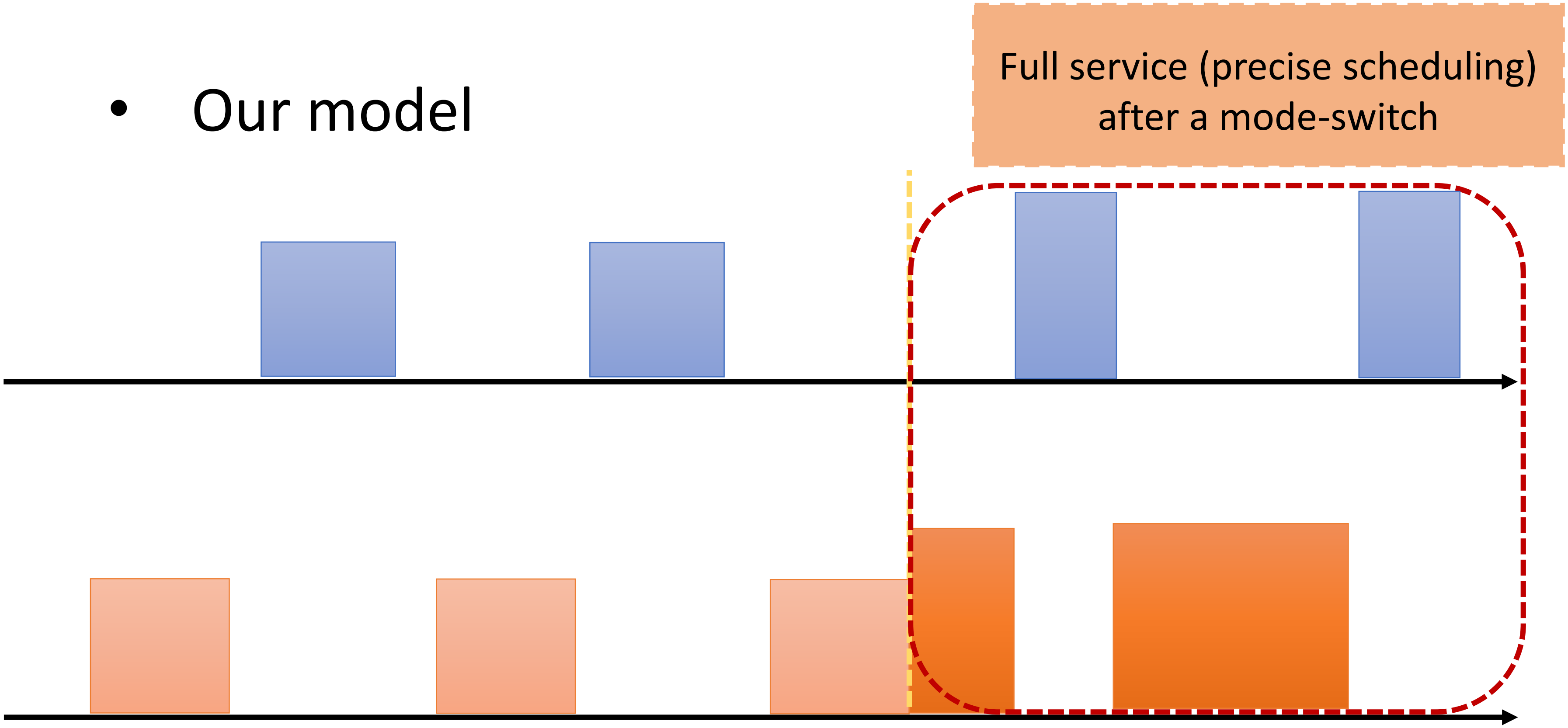
Problem and Motivation

- Traditional MC task model



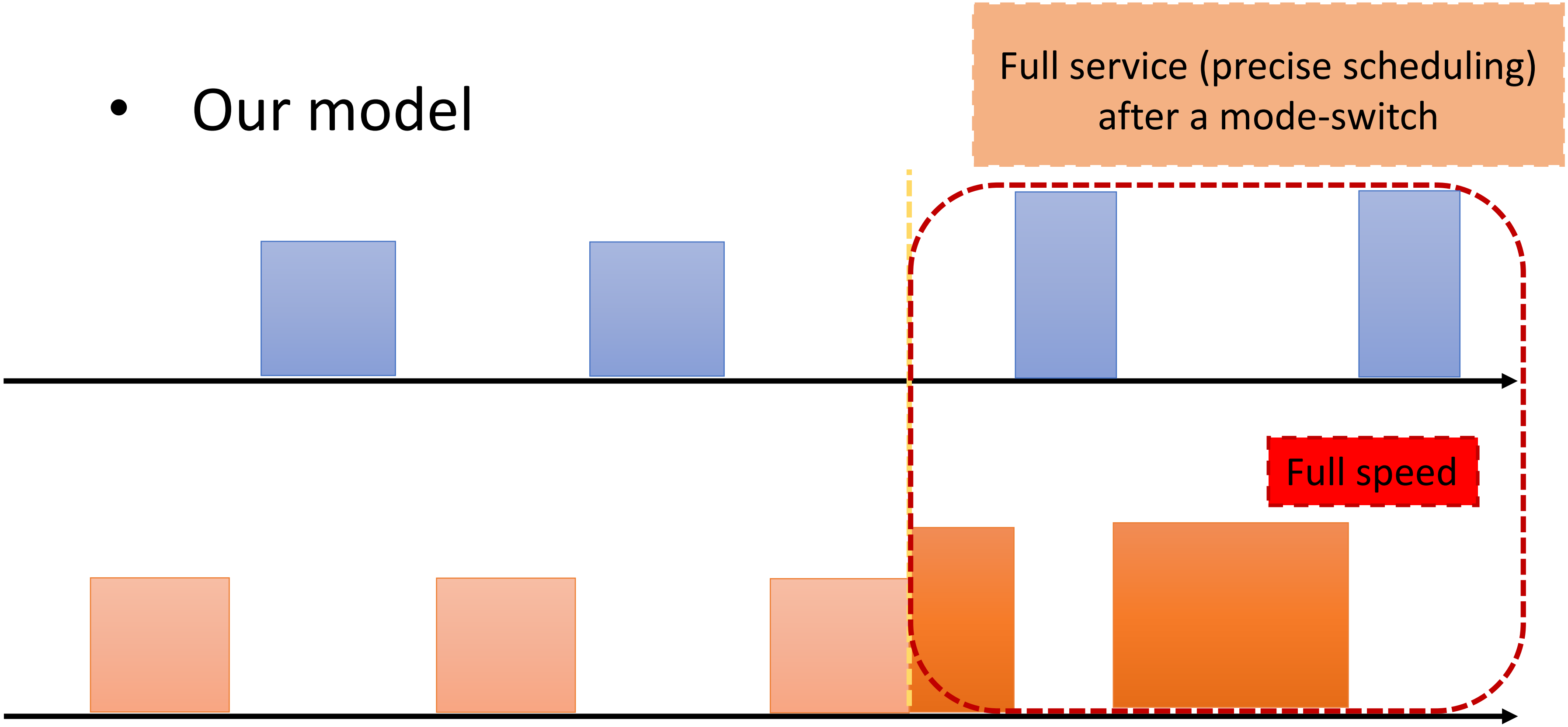
Problem and Motivation

- Our model



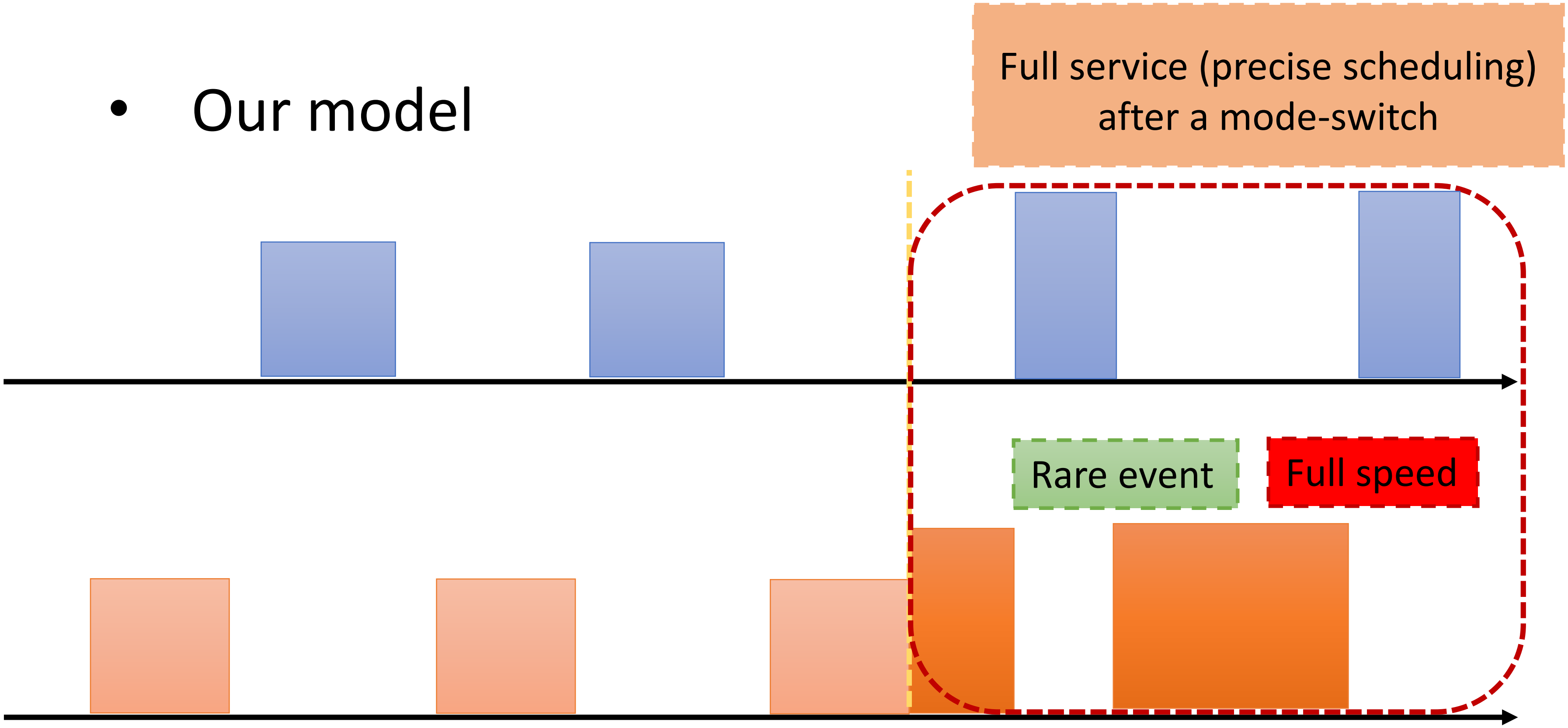
Problem and Motivation

- Our model



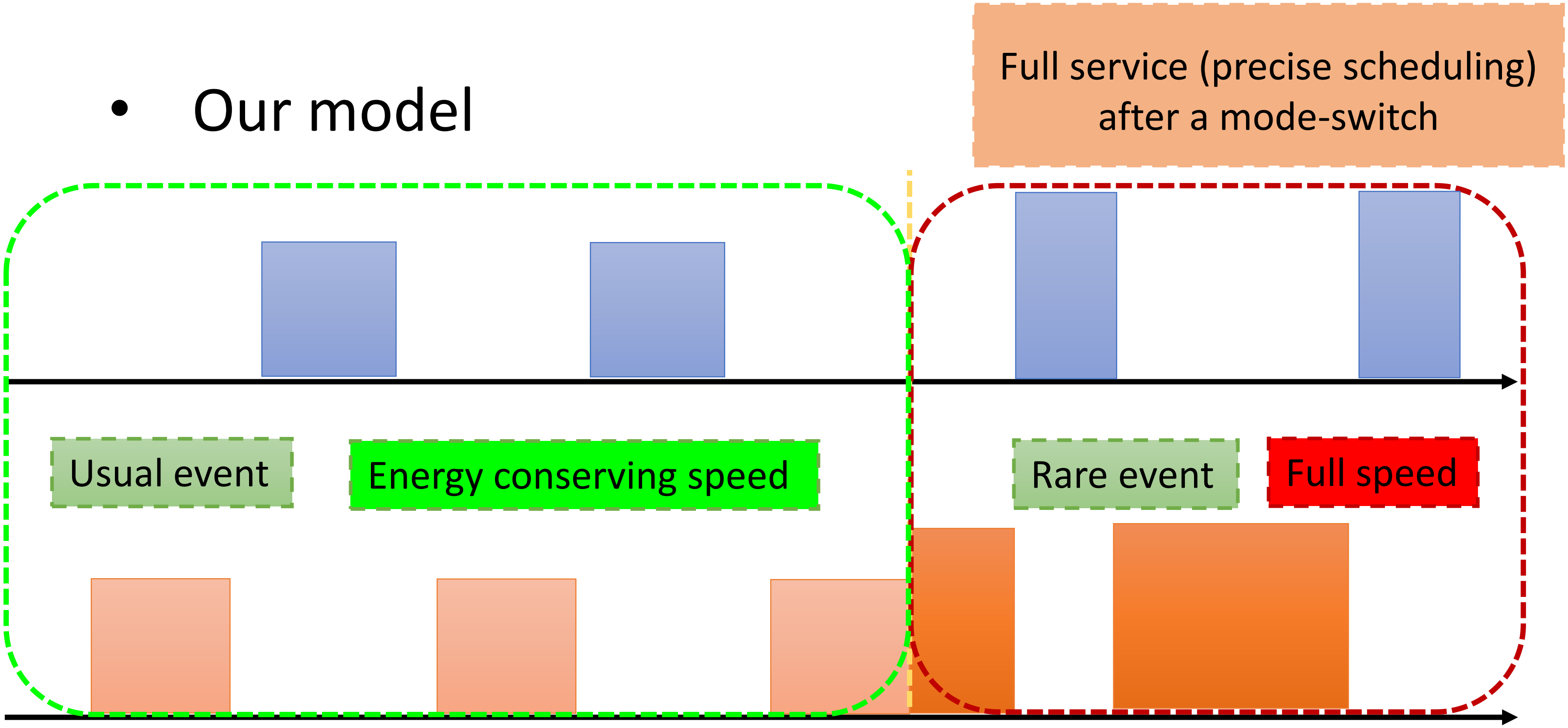
Problem and Motivation

- Our model



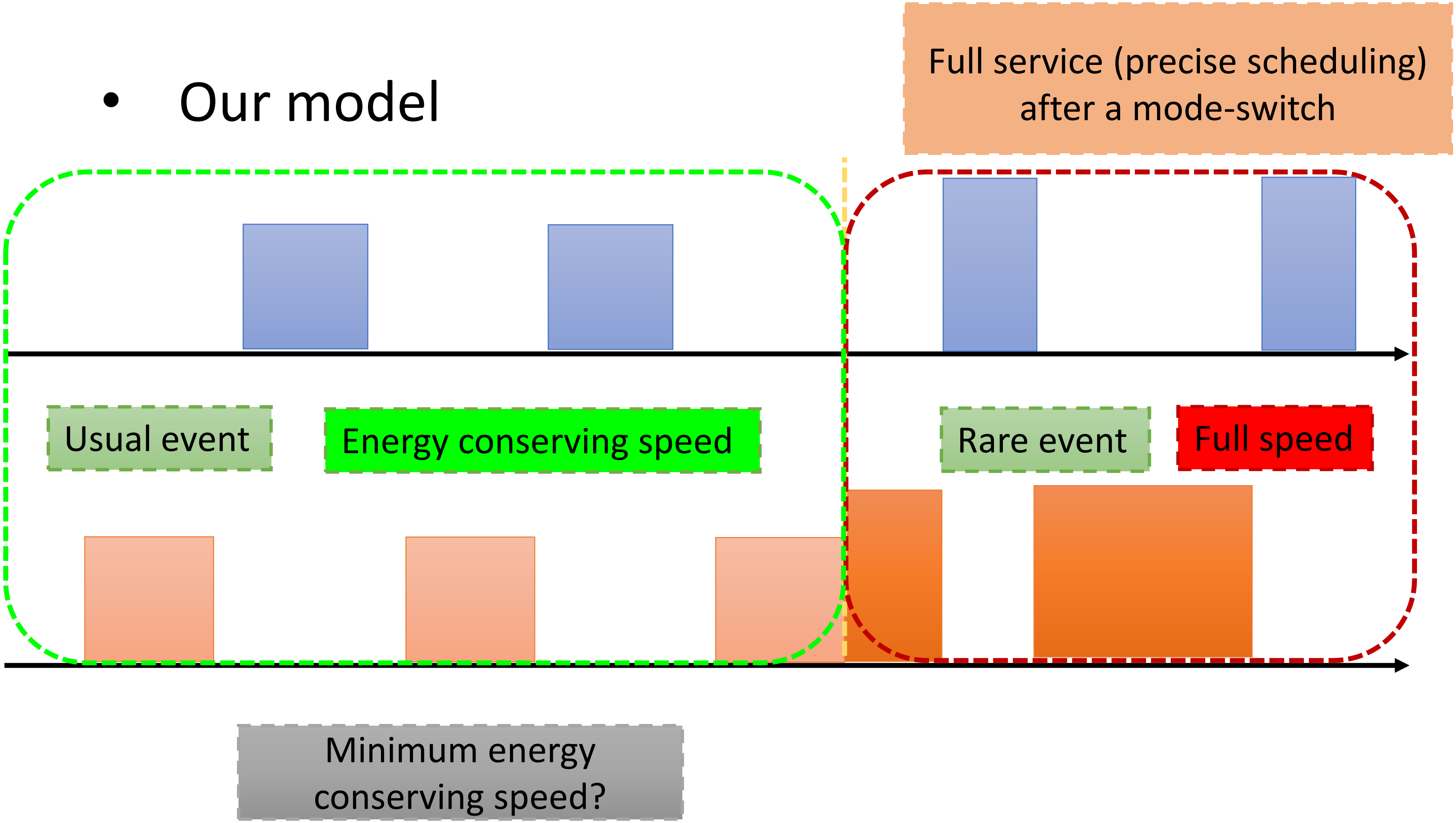
Problem and Motivation

- Our model



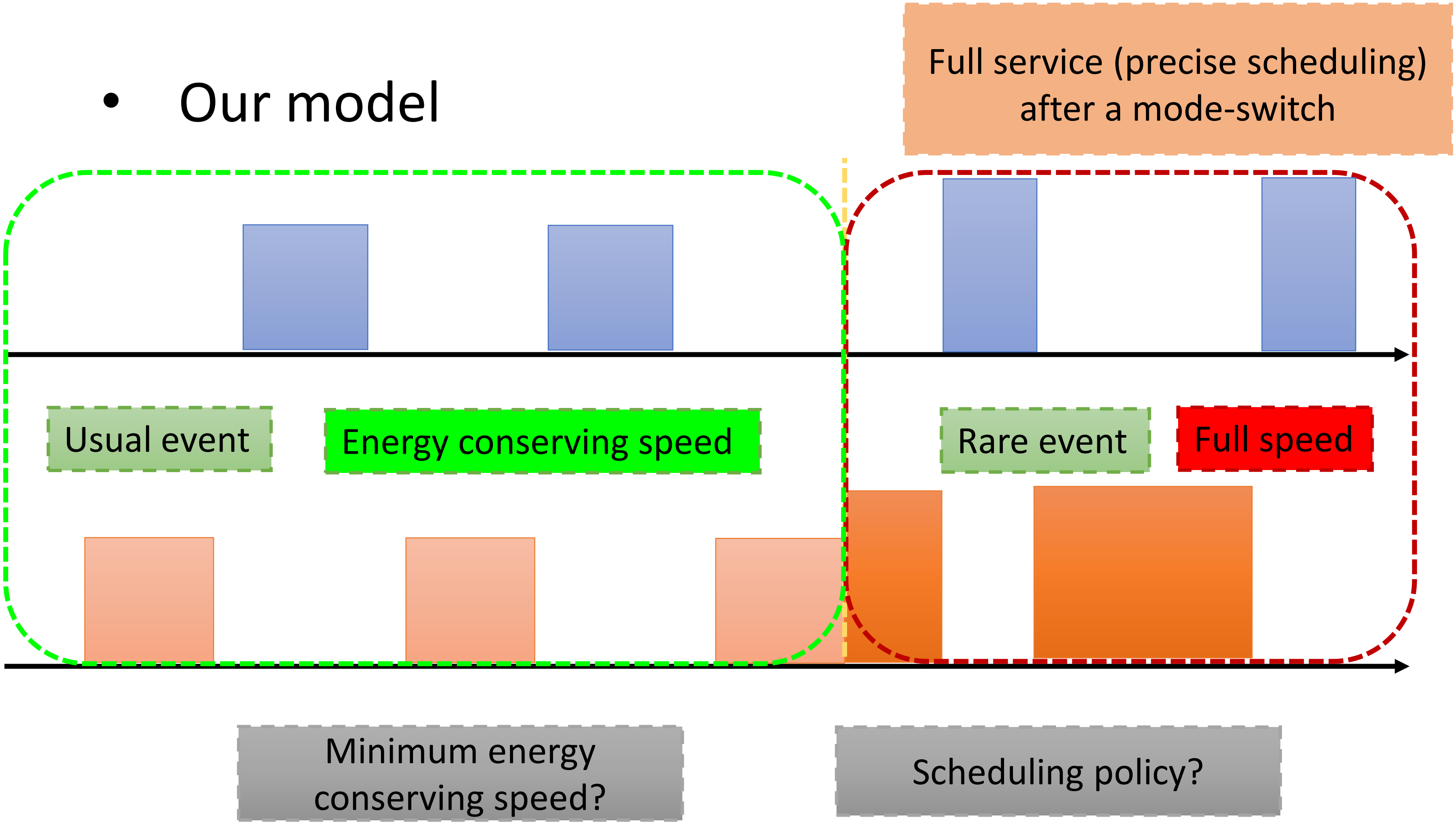
Problem and Motivation

- Our model



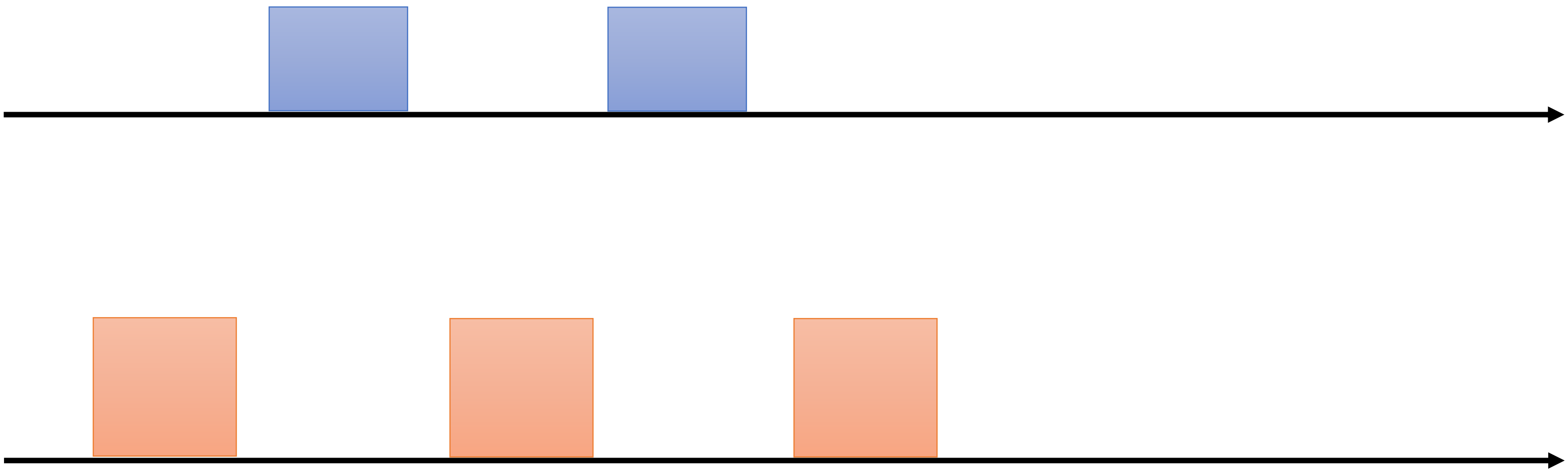
Problem and Motivation

- Our model



EDF-VD

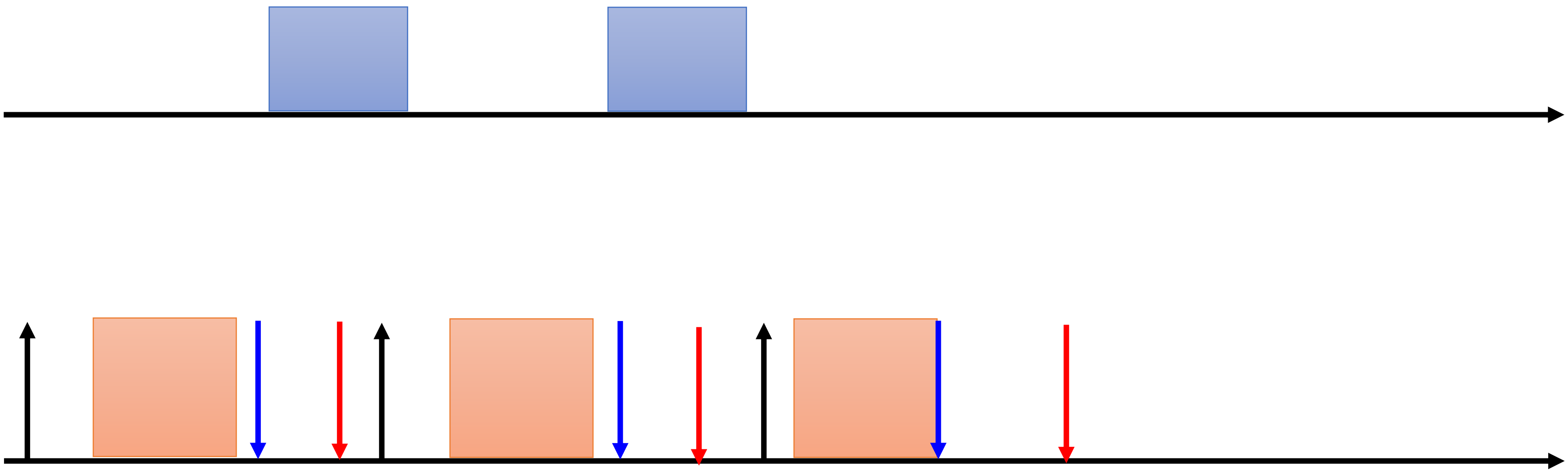
- ❑ Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



S. Baruah, V. Bonifaci, G. DAngelo, H. Li, A. Marchetti-Spaccamela, S. Van Der Ster, and L. Stougie, “The preemptive uniprocessor scheduling of mixed-criticality implicit-deadline sporadic task systems,” in ECRTS 2012

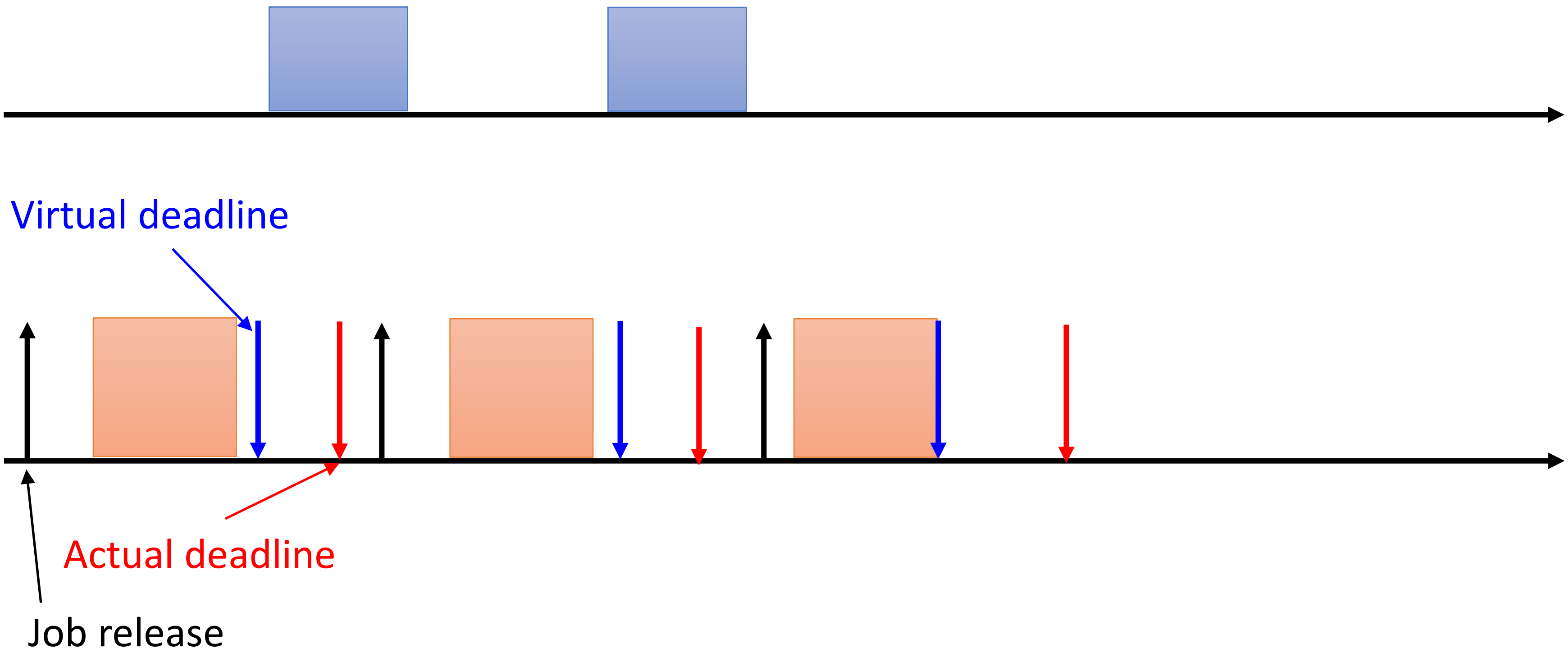
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



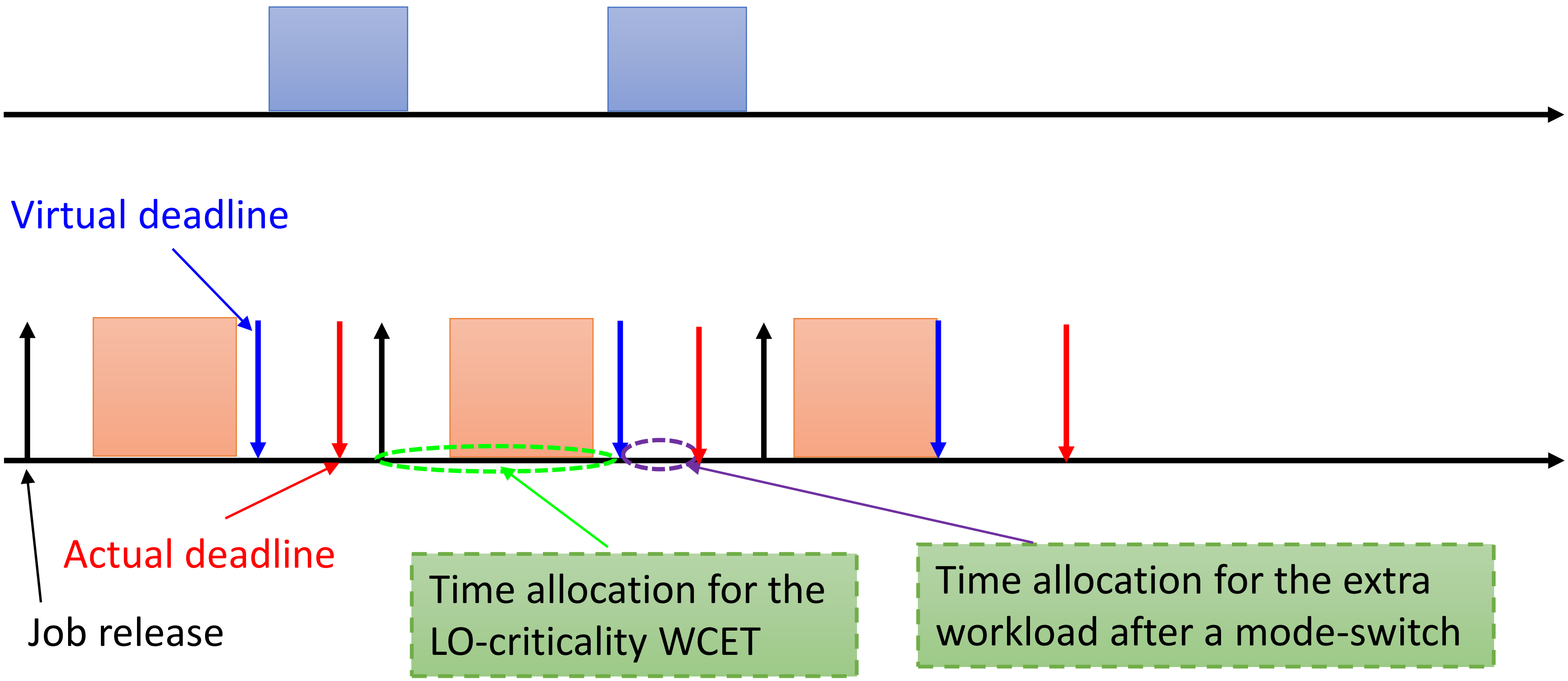
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



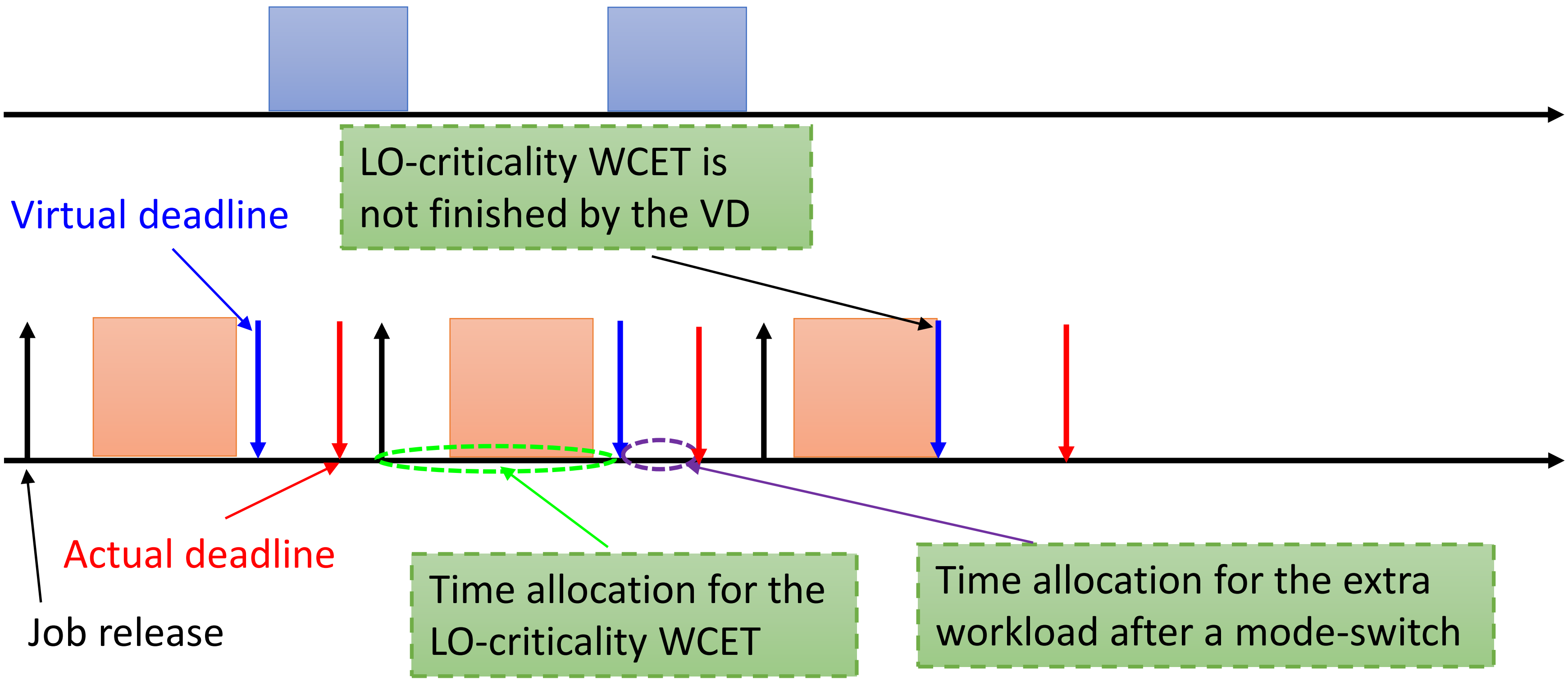
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



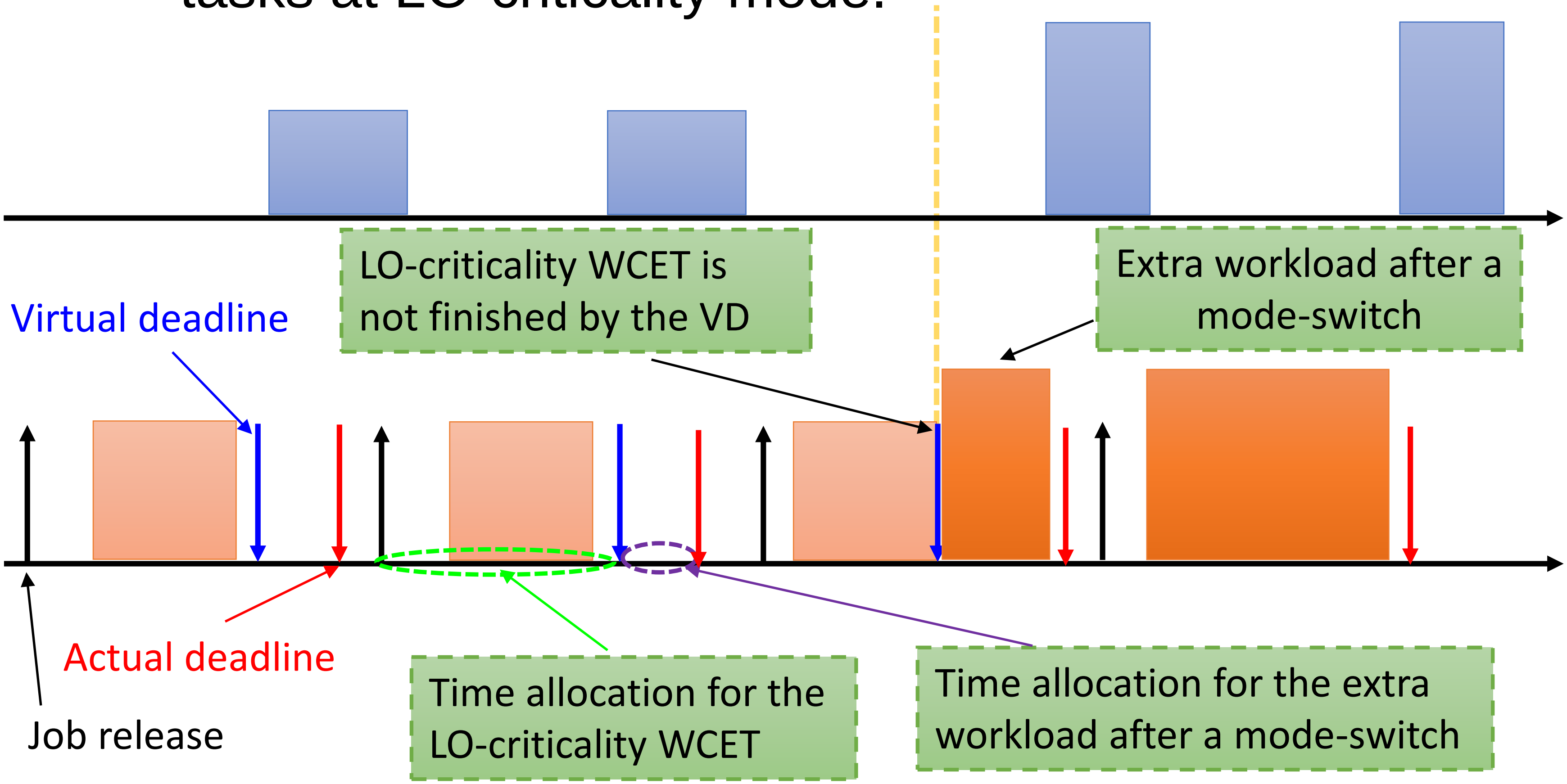
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



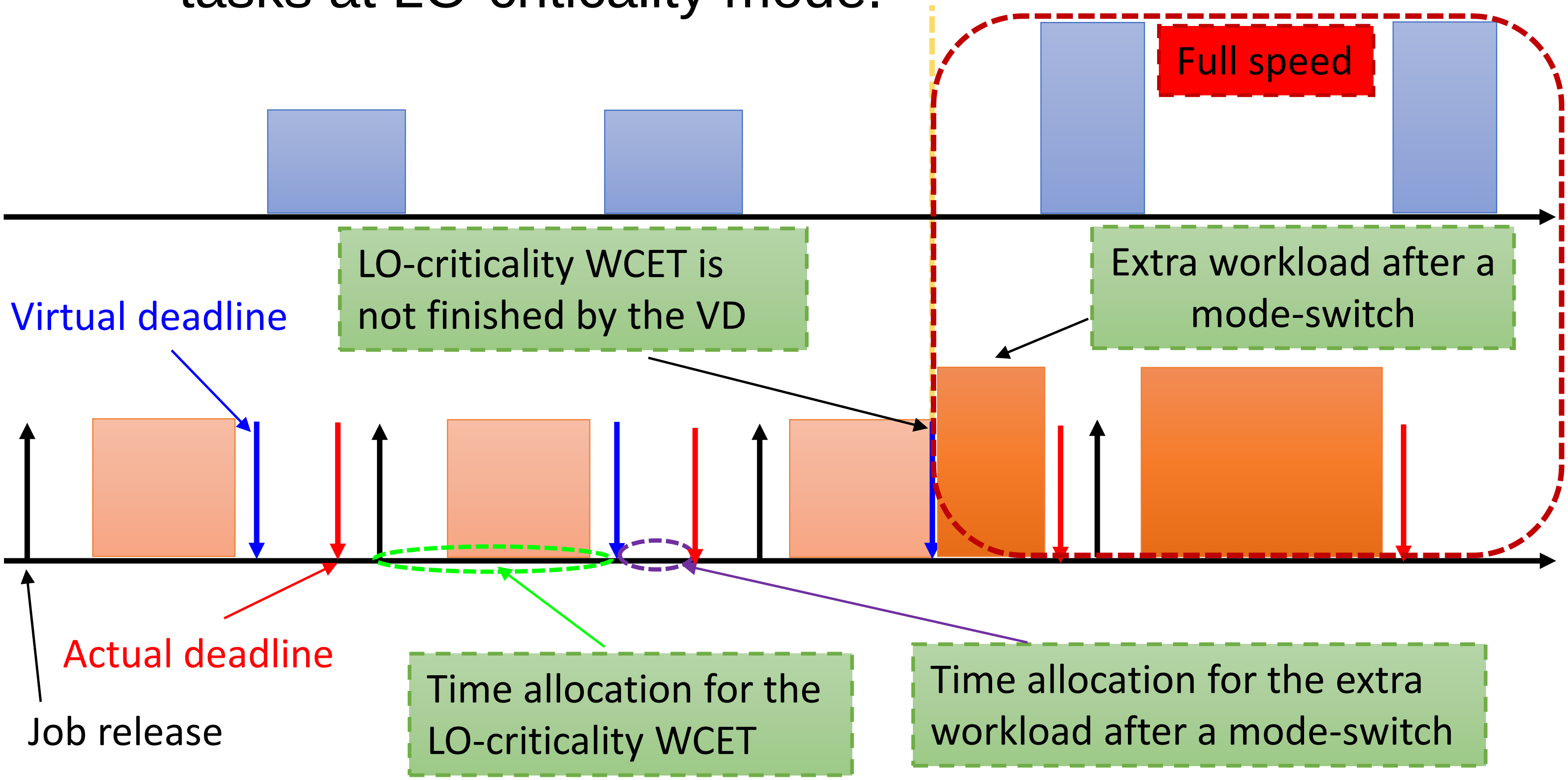
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



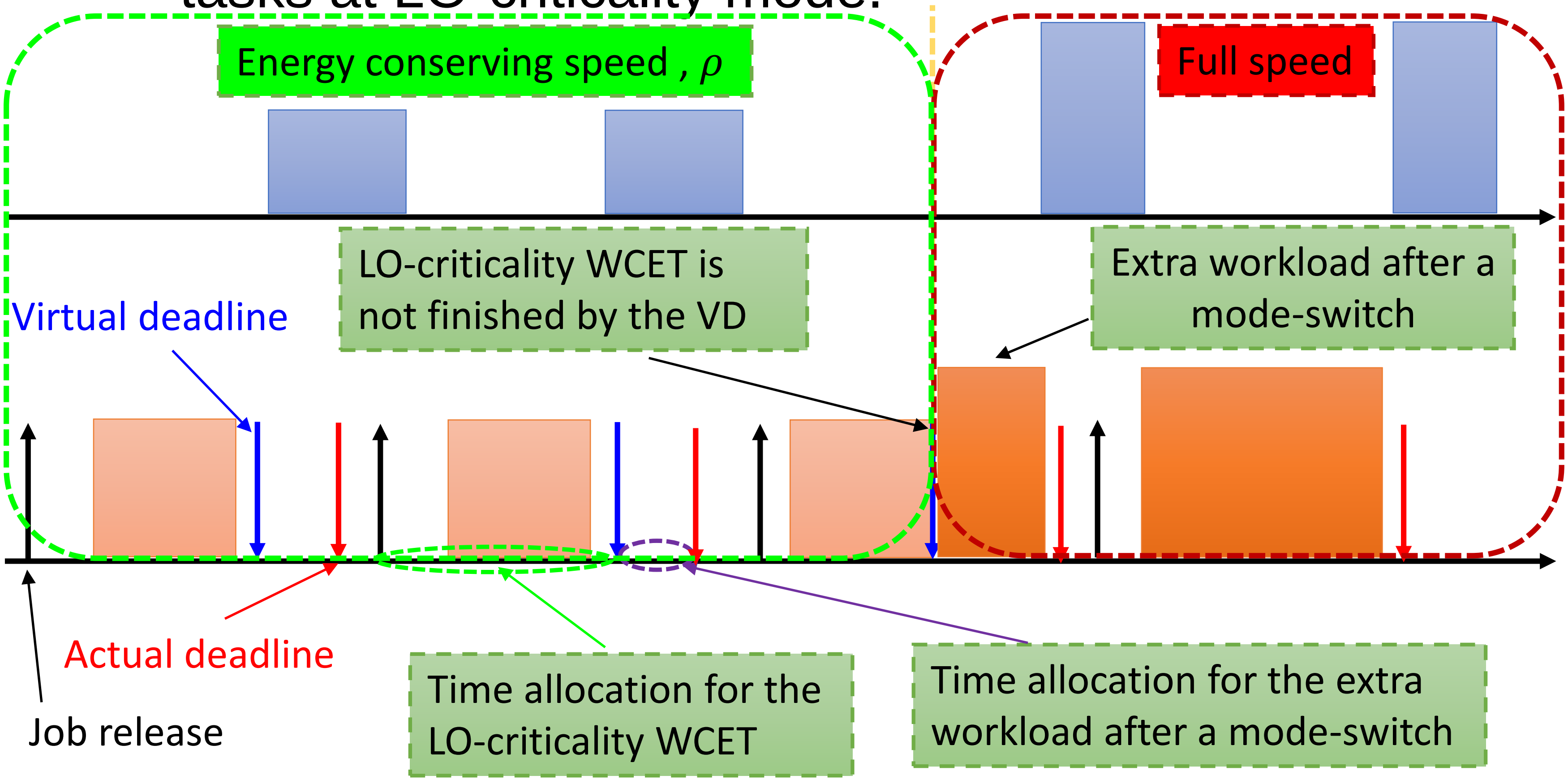
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



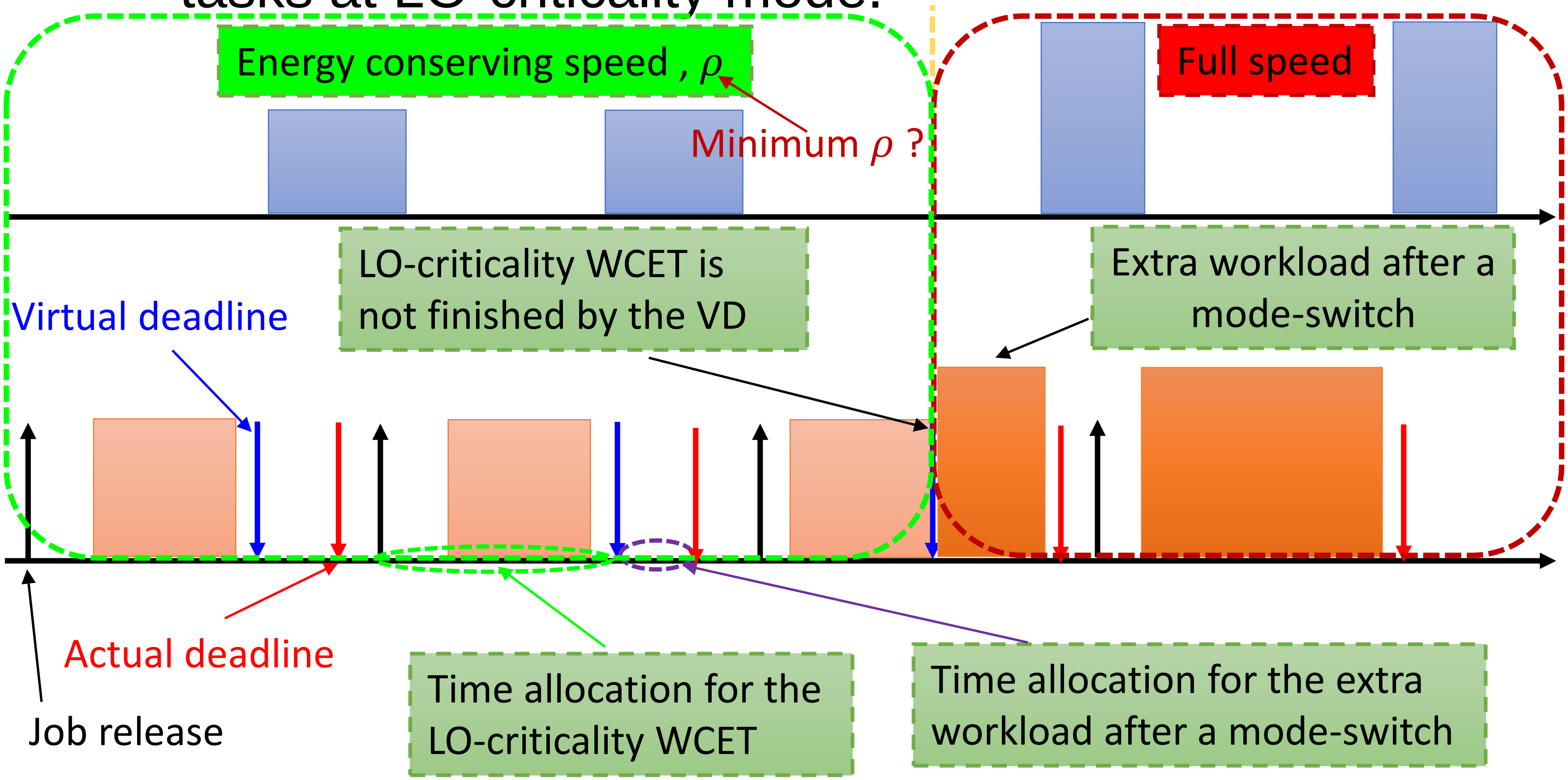
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



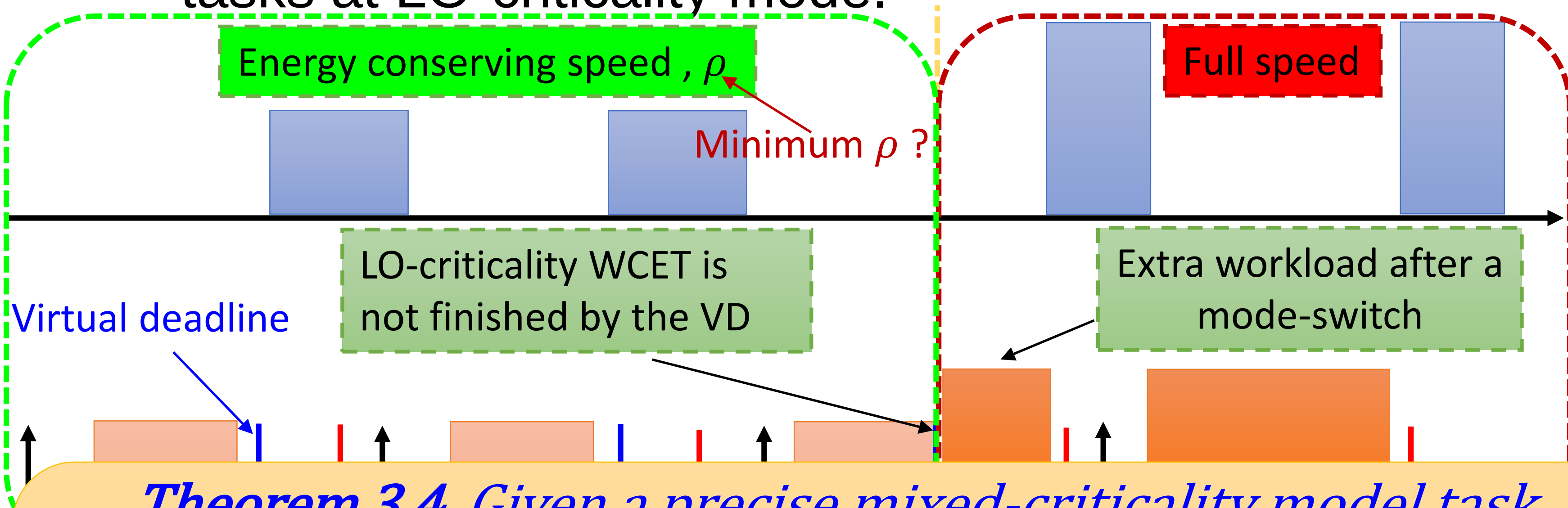
EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.

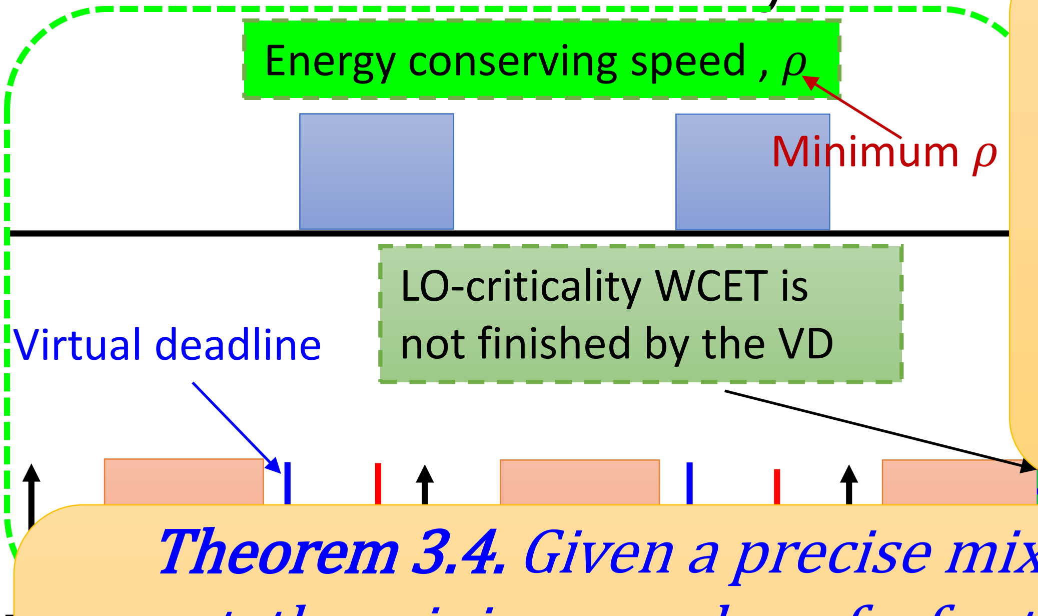


Theorem 3.4. *Given a precise mixed-criticality model task set, the minimum value of ρ for the task set to be schedulable by EDF-VD is*

$$\rho = \min\left(U_L^L + U_H^H, U_L^L + \frac{(1 - U_L^L)U_H^L}{(1 - U_H^H - U_L^L)}\right)$$

EDF-VD

- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



$$Virtual\ Deadline = x \cdot Actual\ Deadline$$

$$x \le \frac{1-(U_H^H + U_L^L)}{(1-U_L^L)}$$

$$x \ge \frac{{U_H}^L}{(\rho - U_L^L)}$$

Theorem 3.4. *Given a precise mixed-criticality model task set, the minimum value of ρ for the task set to be schedulable by EDF-VD is*

$$\rho = \min(U_L^L + U_H^H, U_L^L + \frac{(1-U_L^L)U_H^L}{(1-U_H^H-U_L^L)})$$

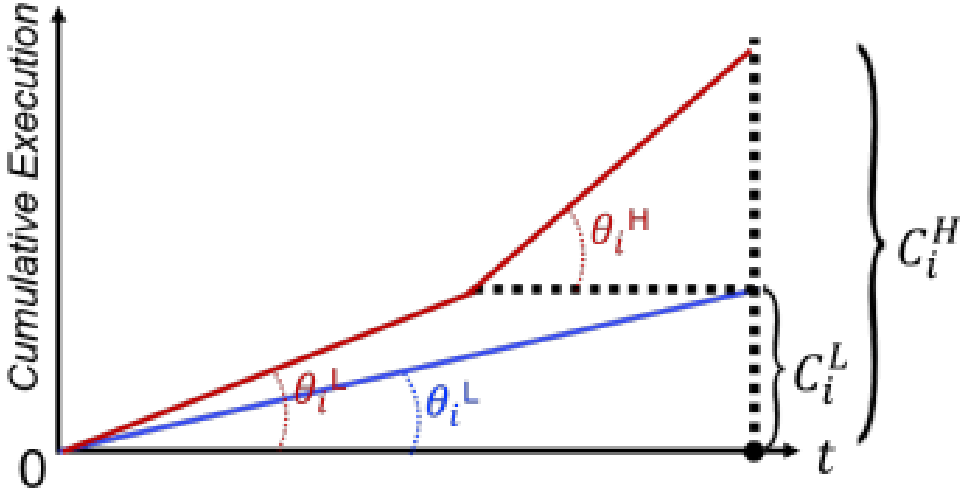
MC-Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.

J. Lee, K. Phan, X. Gu, J. Lee, A. Easwaran, I. Shin, and I. Lee. MC-Fluid: Fluid model-based mixed-criticality scheduling on multiprocessors. In RTSS *2014*.

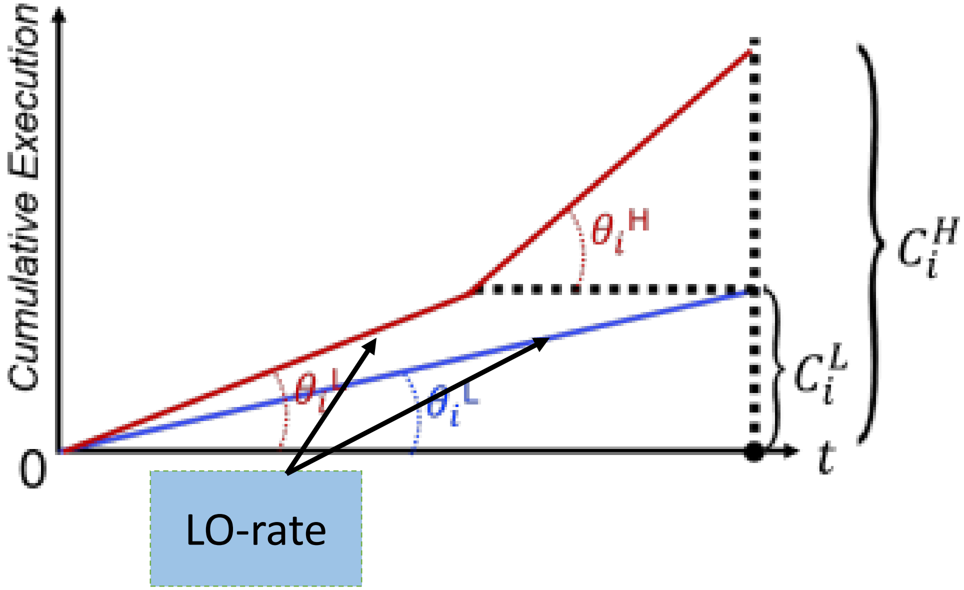
MC-Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



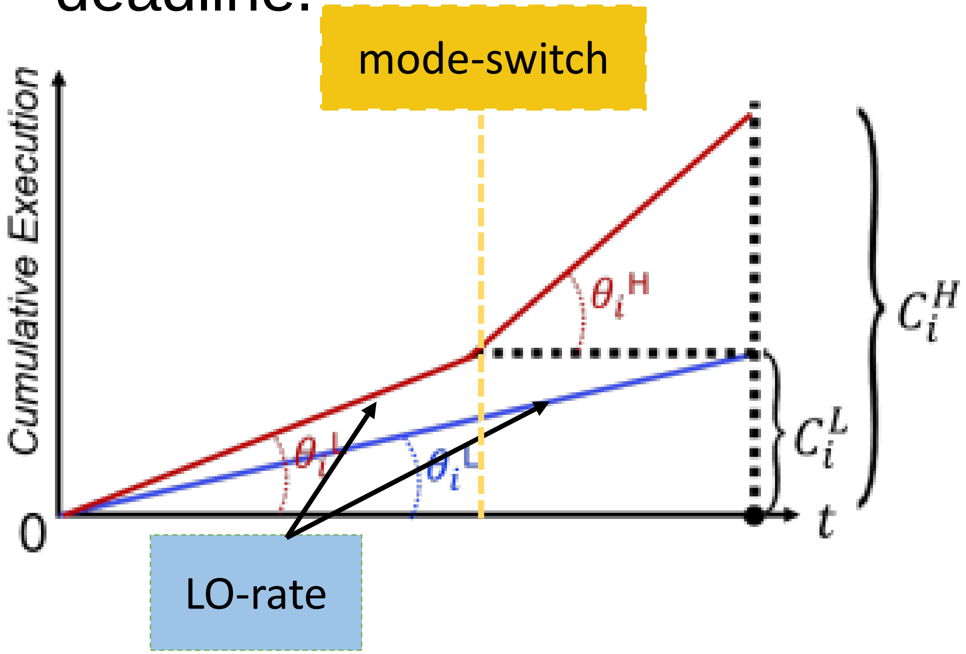
MC-Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



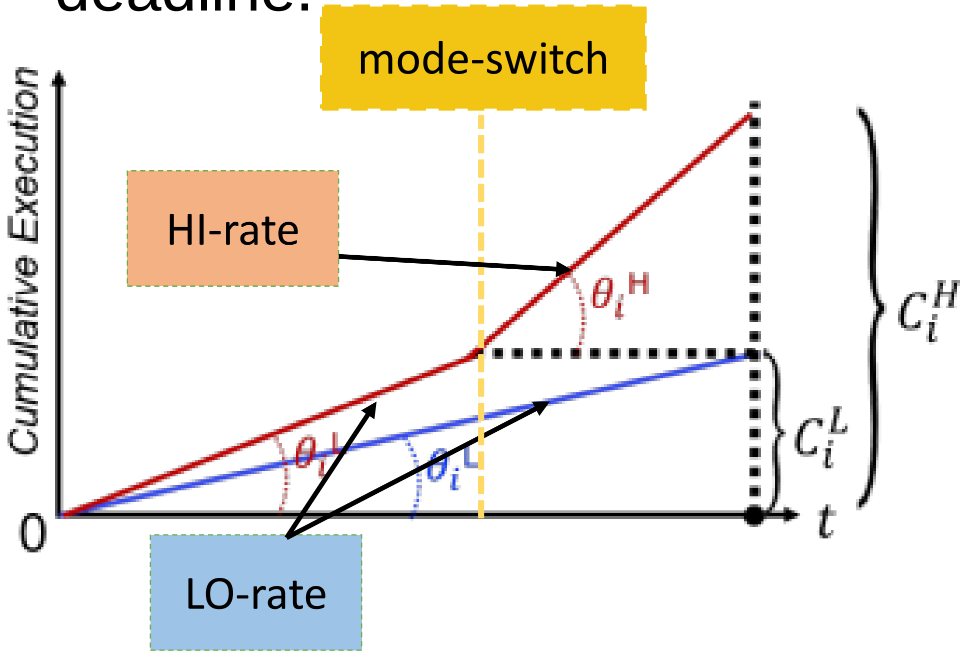
MC-Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



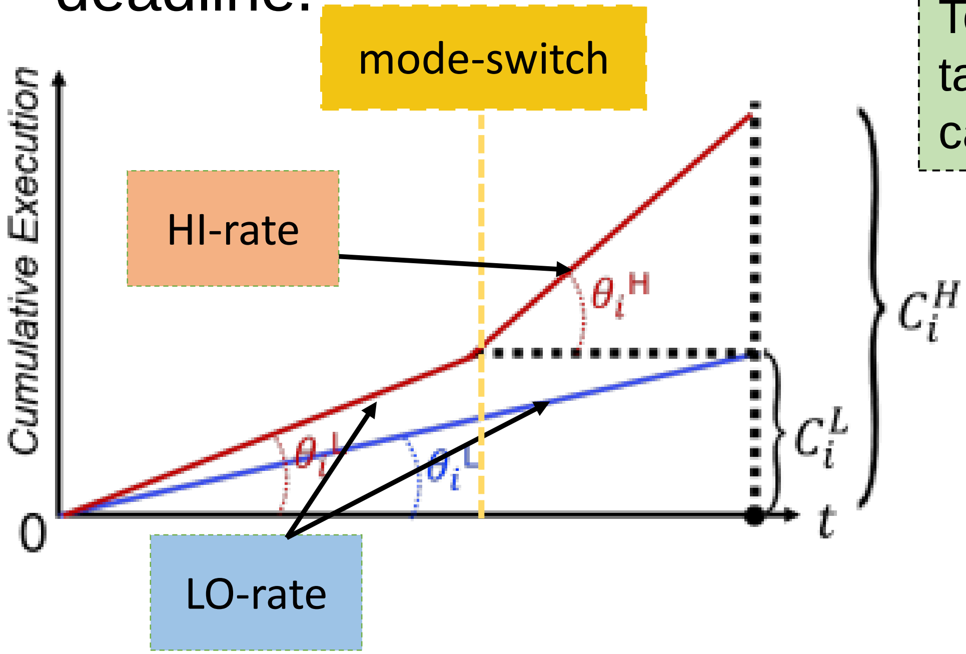
MC-Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



MC-Fluid Scheduling

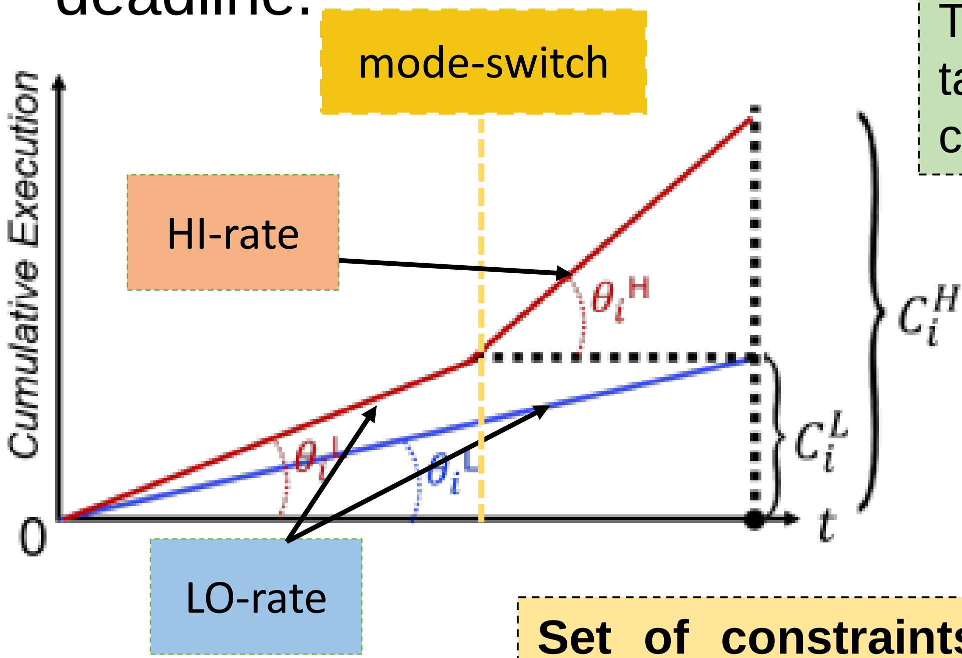
- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



Total processor-share (of each task) is less or equal to the capacity (speed) of the processor.

MC-Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



Total processor-share (of each task) is less or equal to the capacity (speed) of the processor.

Set of constraints to find the rate

$$\sum_{\tau_i \in \tau} \theta_i^{l} < 1 \qquad \sum_{\tau_i \in \tau_H} \theta_i^{H} \leq 1$$

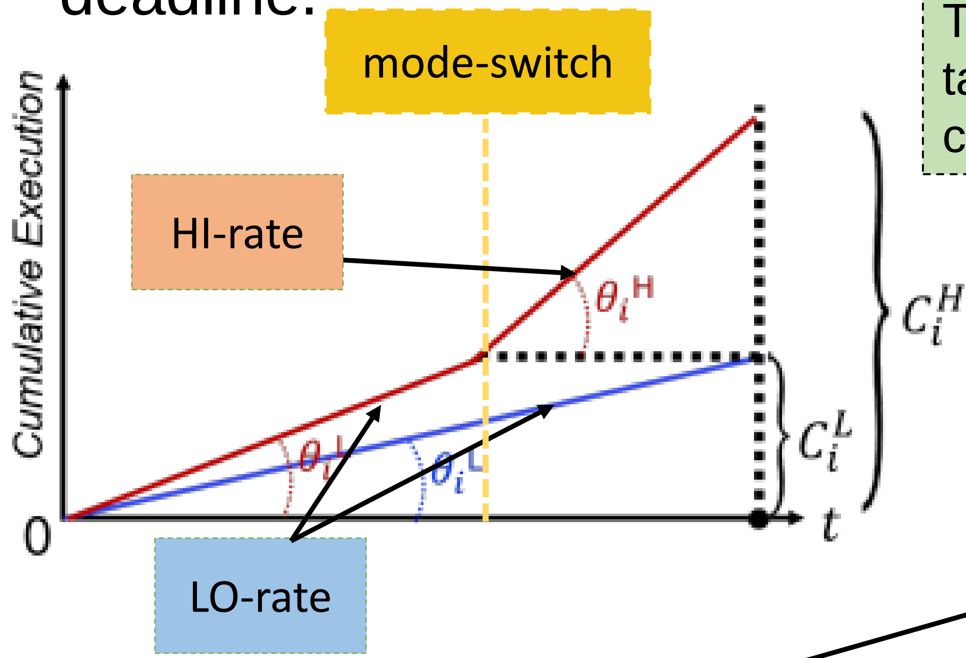
$$\theta_i^{H} = \theta_i \qquad \theta_i^{l} = \lambda \theta_i$$

$$\lambda = \frac{U^L}{1 + U^L - U^H}$$

$$\forall i, \theta_i = \frac{u_i^{L}}{\lambda} - u_i^{L} + u_i^{H}$$

MC-Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



Total processor-share (of each task) is less or equal to the capacity (speed) of the processor.

$$\sum_{\tau_i \in \tau} \theta_i^{\,l} < 1 \qquad \sum_{\tau_i \in \tau_H} \theta_i^{\,H} \leq 1$$

$$\theta_i^{\,H} = \theta_i \qquad \theta_i^{\,l} = \lambda\theta_i$$

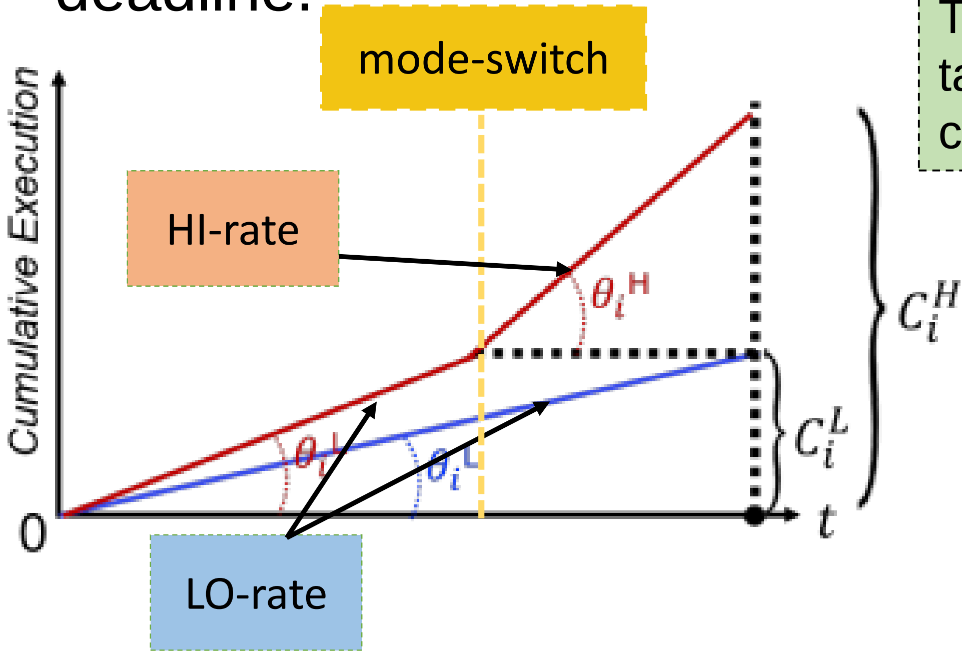
$$\lambda = \frac{U^L}{1 + U^L - U^H}$$

$$\forall i,\ \theta_i = \frac{u_i^{\,L}}{\lambda} - u_i^{\,L} + u_i^{\,H}$$

The ratio of Hi and LO rate is a constant

Dual Rate Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.

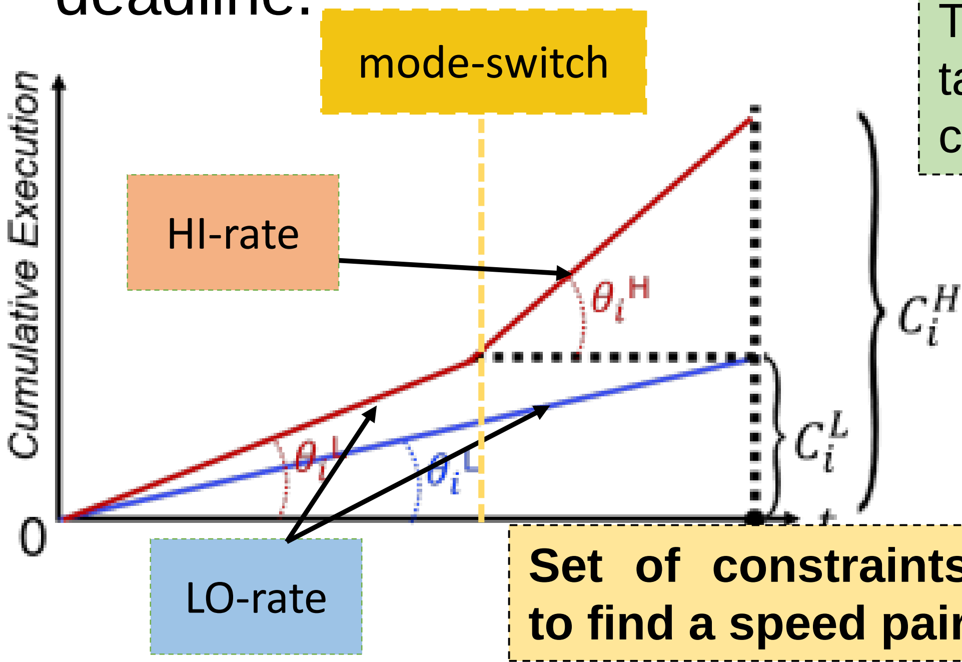


Total processor-share (of each task) is less or equal to the capacity (speed) of the processor.

Relaxed this constraint, and find dual rate for each task

Dual Rate Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



Total processor-share (of each task) is less or equal to the capacity (speed) of the processor.

$$\sum_i \theta_i^l \leq \rho \qquad \sum_i \theta_i^H \leq 1$$

$$\frac{C_i^l}{\theta_i^l} \leq T_i \qquad \frac{C_i^H}{\theta_i^H} \leq T_i$$

$$\theta_i^l \leq \theta_i^H$$

$$\frac{C_i^l}{\theta_i^l} + \frac{C_i^H - C_i^l}{\theta_i^H} \leq T_i$$

Set of constraints to find a speed pair

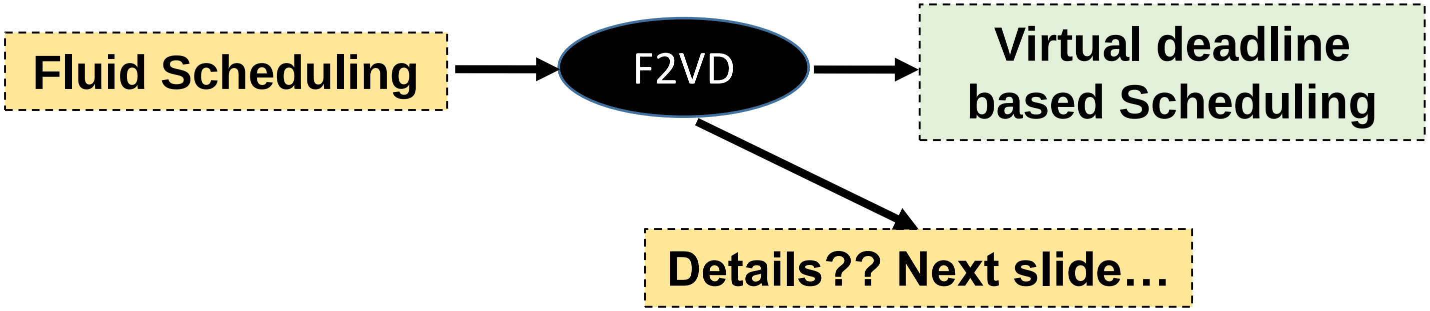
Relaxed this constraint, and find dual rate for each task

Dual Rate Fluid Scheduling

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.
- Fluid scheduling is not practical to implement due to runtime overheads and frequent context switches.

Fluid to Virtual Deadline: F2VD

- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.
- Fluid scheduling is not practical to implement due to runtime overheads and frequent context switches.



Fluid to Virtual Deadline: F2VD

LO-criticality mode

Find the pair (θ_i^l, θ_i^H)
for all tasks

Fluid to Virtual Deadline: F2VD

LO-criticality mode

Find the pair (θ_i^l, θ_i^H)
for all tasks

The pair must satisfy these constraints

$$\sum_i \theta_i^l \le \rho \qquad \sum_i \theta_i^H \le 1$$

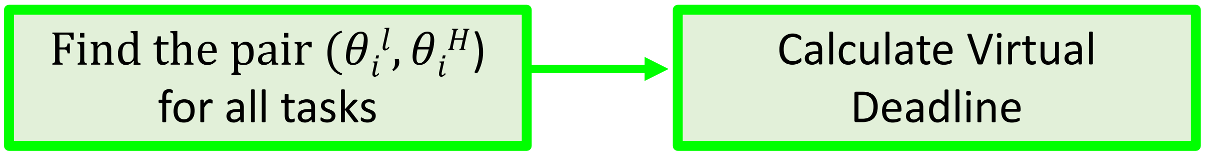
$$\frac{C_i^l}{\theta_i^l} \le T_i \qquad \frac{C_i^H}{\theta_i^H} \le T_i$$

$$\theta_i^l \le \theta_i^H$$

$$\frac{C_i^l}{\theta_i^l} + \frac{C_i^H - C_i^l}{\theta_i^H} \le T_i$$

Fluid to Virtual Deadline: F2VD

LO-criticality mode



Fluid to Virtual Deadline: F2VD

LO-criticality mode

Find the pair $(\theta_i^{l}, \theta_i^{H})$ for all tasks → Calculate Virtual Deadline → Processor operates at the degraded speed and schedule jobs by EDF according to their virtual deadline.

Fluid to Virtual Deadline: F2VD

LO-criticality mode

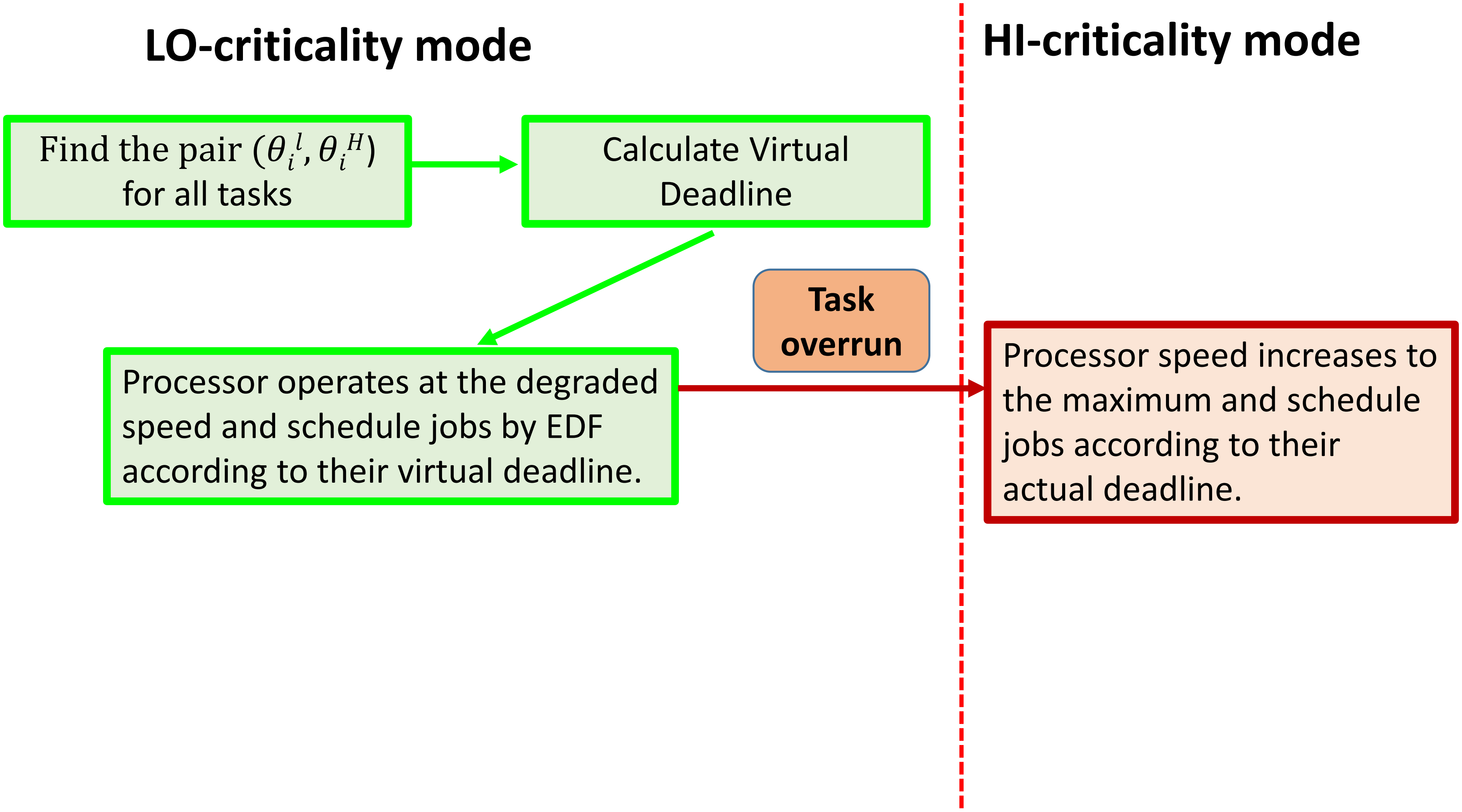
Find the pair $(\theta_i^{l}, \theta_i^{H})$ for all tasks

Calculate Virtual Deadline

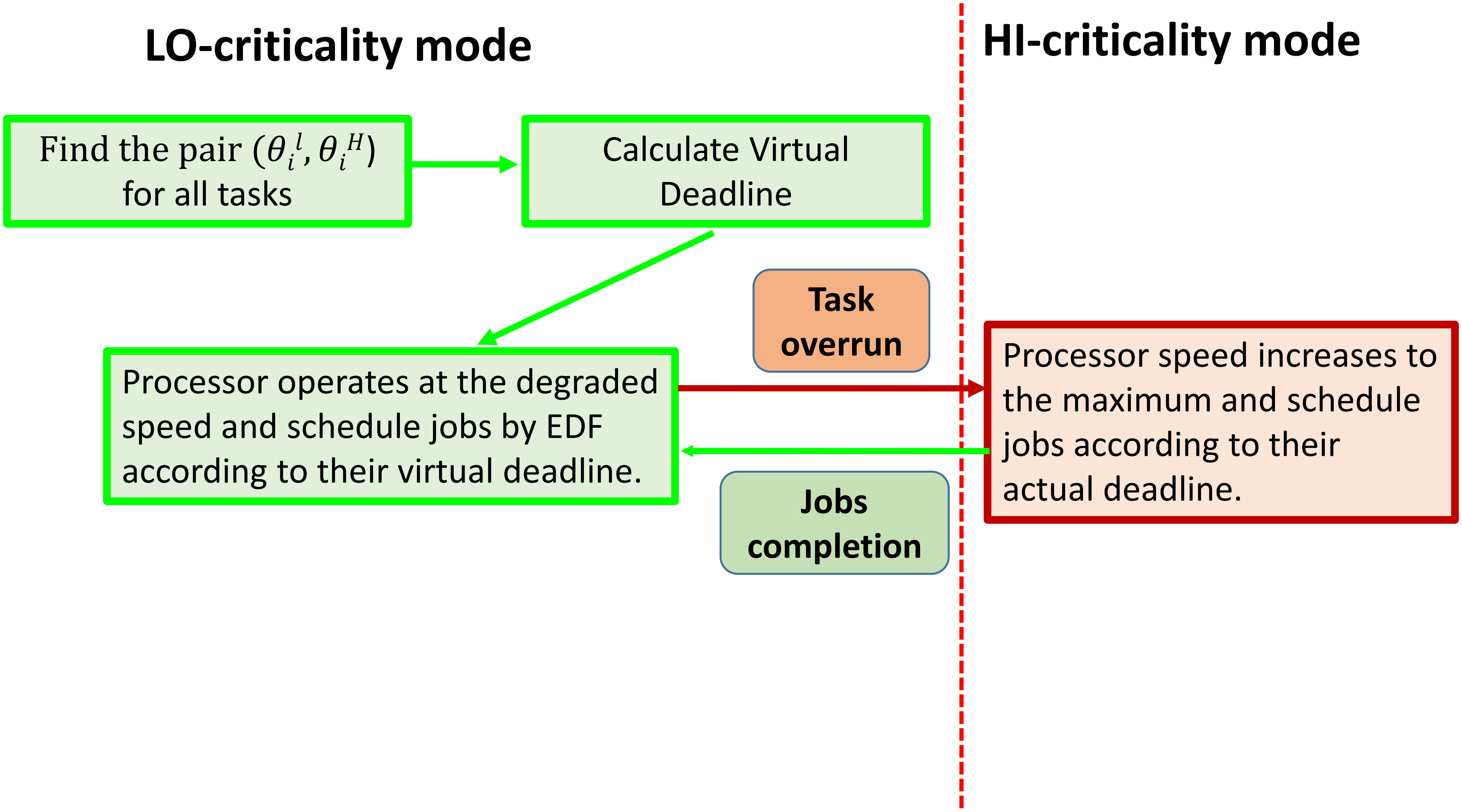
Processor operates at the degraded speed and schedule jobs by EDF according to their virtual deadline.

Task overrun

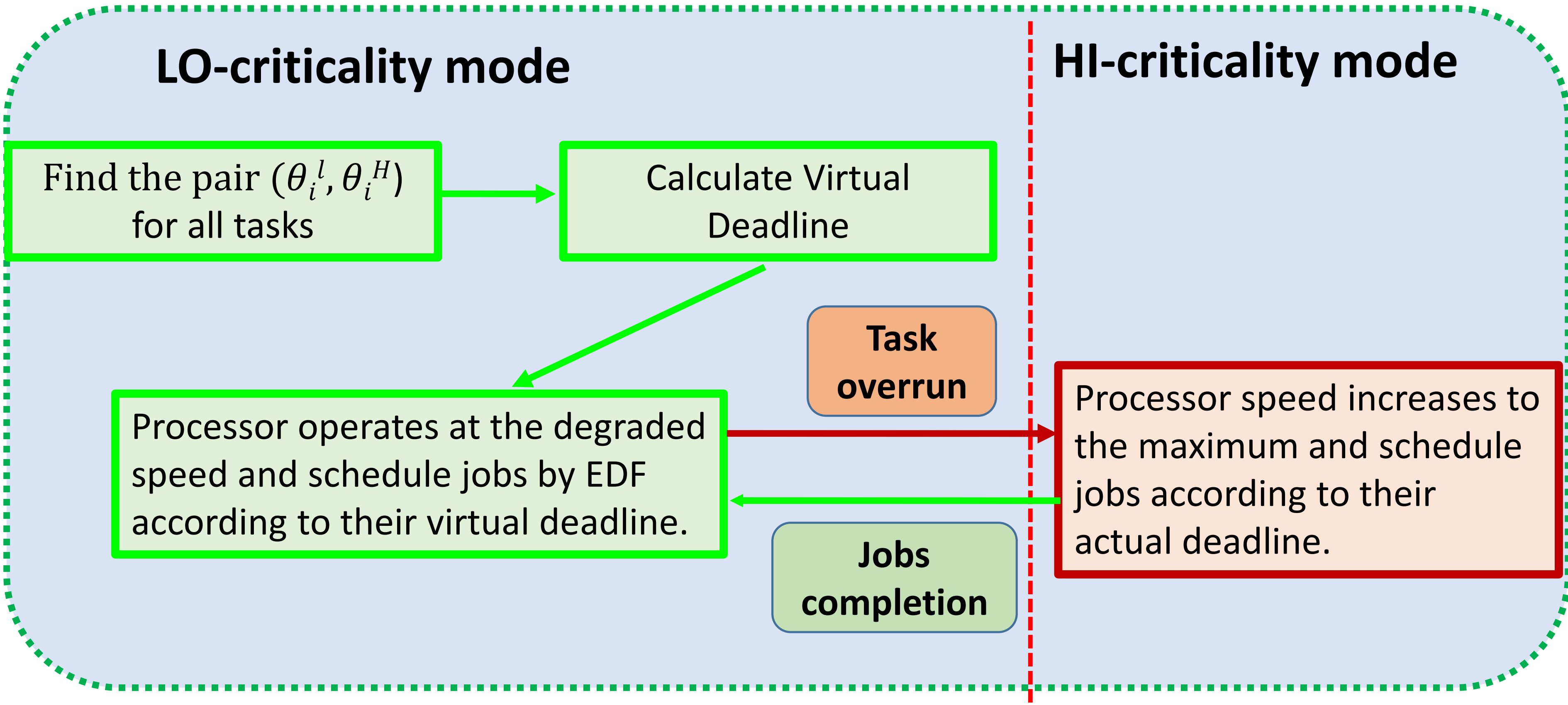
Fluid to Virtual Deadline: F2VD



Fluid to Virtual Deadline: F2VD



Fluid to Virtual Deadline: F2VD



No schedulability loss

Exploiting Probabilistic Information

- So far, we have assumed all tasks execute up to its WCET at the respective criticality-level.

Exploiting Probabilistic Information

- So far, we have assumed all tasks execute up to its WCET at the respective criticality-level.
- In Practice, a task rarely needs to execute up to its WCET.

Exploiting Probabilistic Information

- So far, we have assumed all tasks execute up to its WCET at the respective criticality-level.
- In Practice, a task rarely needs to execute up to its WCET.
- Exploit the probabilistic based prediction strategy and the DVFS scheme to the precise scheduling of MC tasks.

Approach Overview

Probabilistic analysis

From the probabilistic execution time, find the optimal $p^{LO->HI}$

Response-Time analysis

Approach Overview

UCF

Probabilistic analysis

From the probabilistic execution time, find the optimal $p^{LO\text{->}HI}$

↓

Compute C^{LO} from $p^{LO\text{->}HI}$

Response-Time analysis

Approach Overview

UCF

Probabilistic analysis

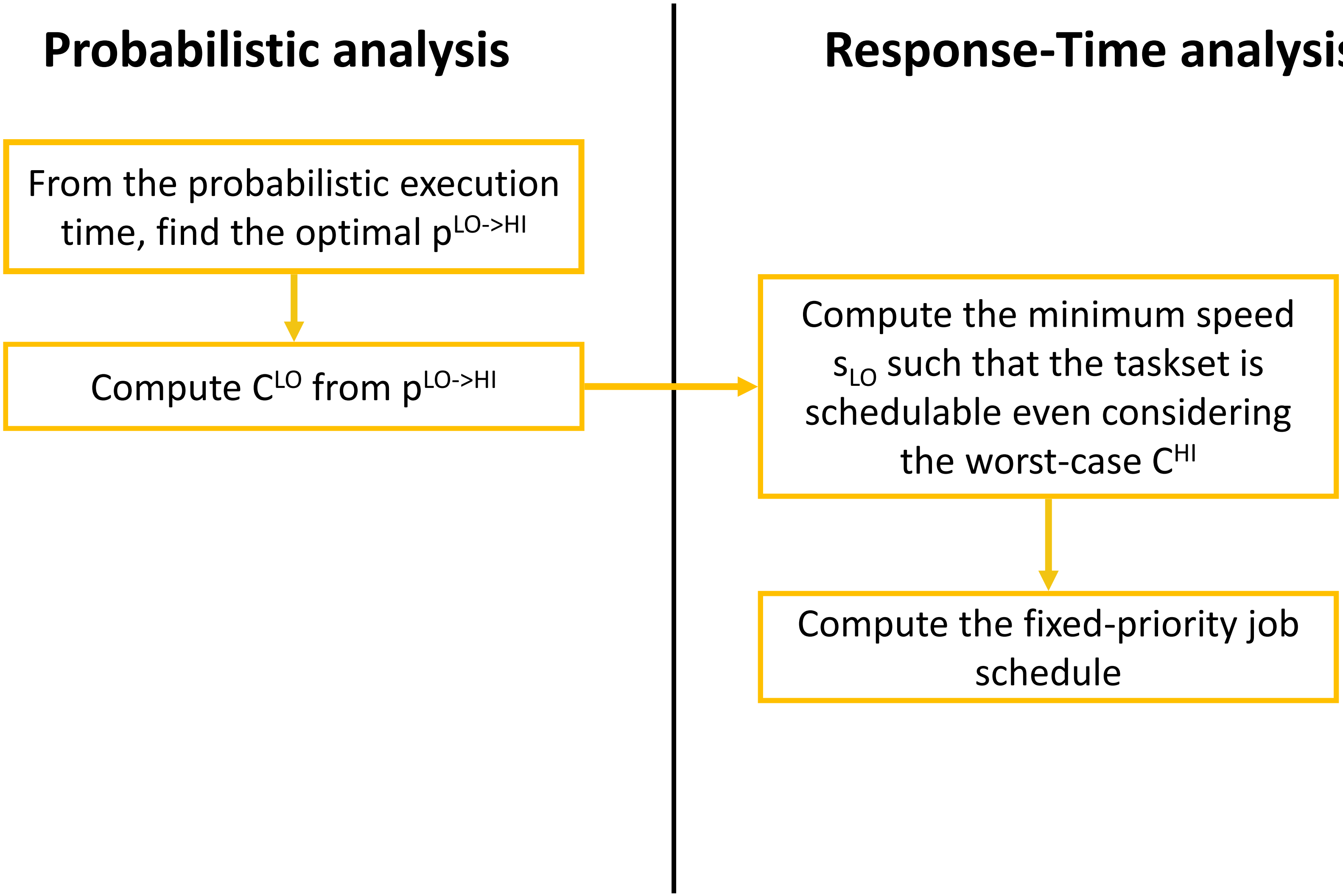
From the probabilistic execution time, find the optimal $p^{LO\text{->}HI}$

Compute C^{LO} from $p^{LO\text{->}HI}$

Response-Time analysis

Compute the minimum speed s_{LO} such that the taskset is schedulable even considering the worst-case C^{HI}

Approach Overview



Approach Overview

Probabilistic analysis

From the probabilistic execution time, find the optimal $p^{LO\text{->}HI}$

Compute C^{LO} from $p^{LO\text{->}HI}$

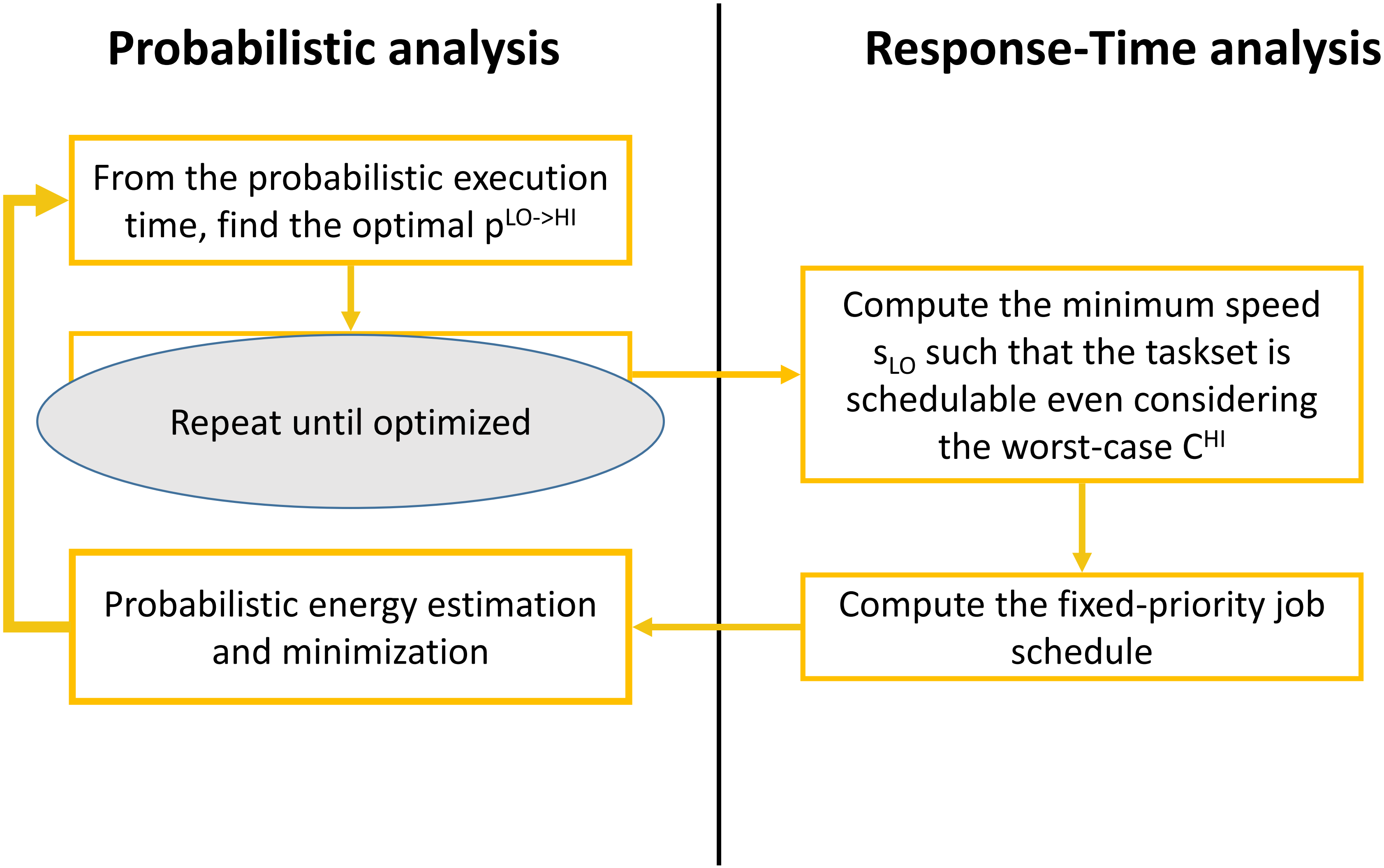
Probabilistic energy estimation and minimization

Response-Time analysis

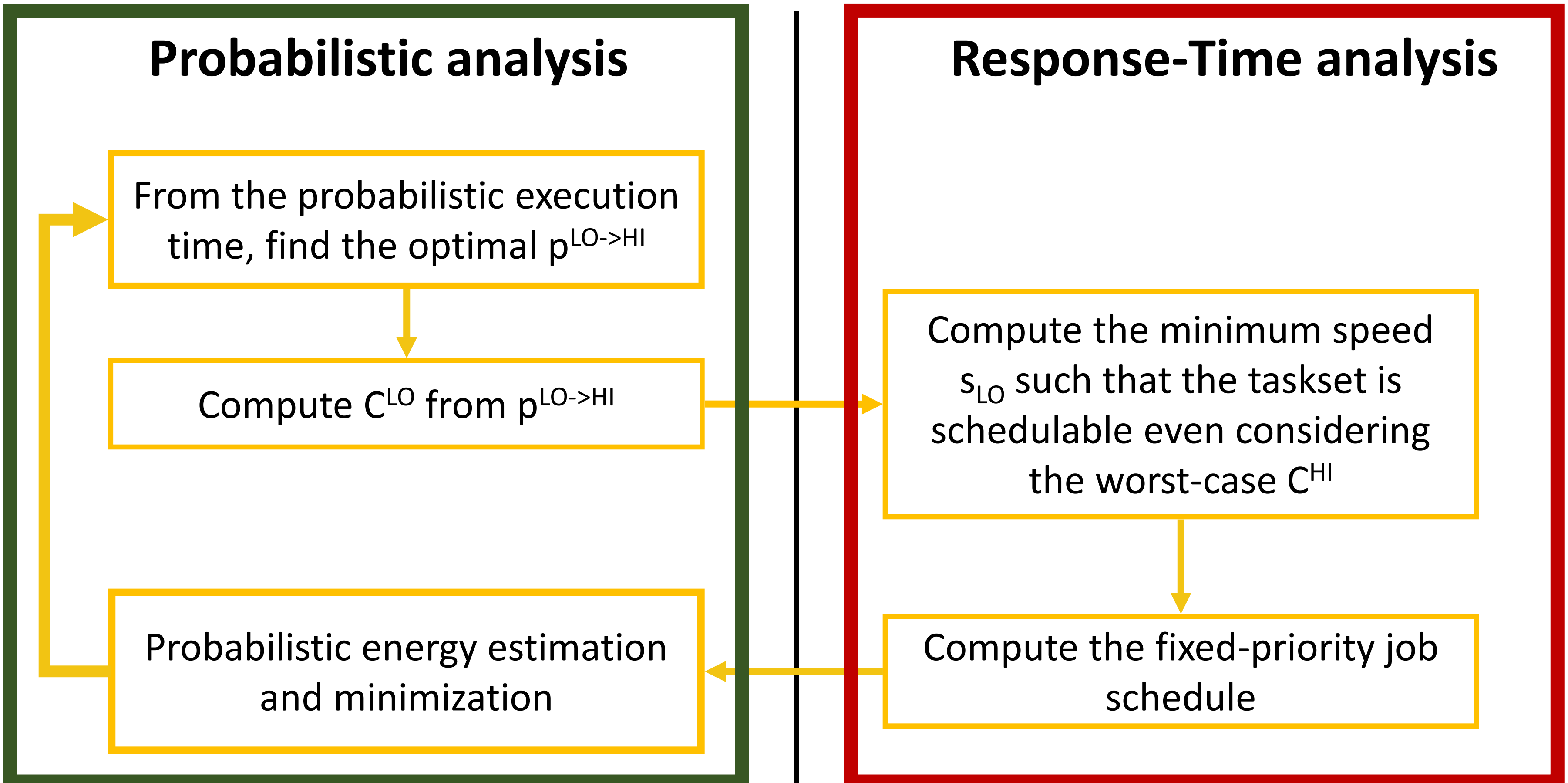
Compute the minimum speed s_{LO} such that the taskset is schedulable even considering the worst-case C^{HI}

Compute the fixed-priority job schedule

Approach Overview



Approach Overview



Energy optimization based on pET → non-pessimistic

Schedulability based on WCET → safe

Conclusion

- In this Presentation, we have covered the following
  - Precise scheduling of MC task using the EDF-VD and MC-Fluid scheduling

Conclusion

- In this Presentation, we have covered the following
  - Precise scheduling of MC task using the EDF-VD and MC-Fluid scheduling
  - Dual flat rate scheduling and its transformation to EDF-VD Family

Conclusion

- In this Presentation, we have covered the following
  - Precise scheduling of MC task using the EDF-VD and MC-Fluid scheduling
  - Dual flat rate scheduling and its transformation to EDF-VD Family
  - Probabilistic based prediction strategy to minimize energy consumption

Acknowledgements

UCF

Some of the slides and figures are adopted from our RTNS'19, ICCAD'20, and EMSOFT'20.

Thank You