Precise Scheduling of Mixed-Criticality Tasks by Varying Processor Speed

Ashikahmed Bhuiyan[1*], **Zhishan Guo[1]**, Sai Sruti[2], and Kecheng Yang[3]

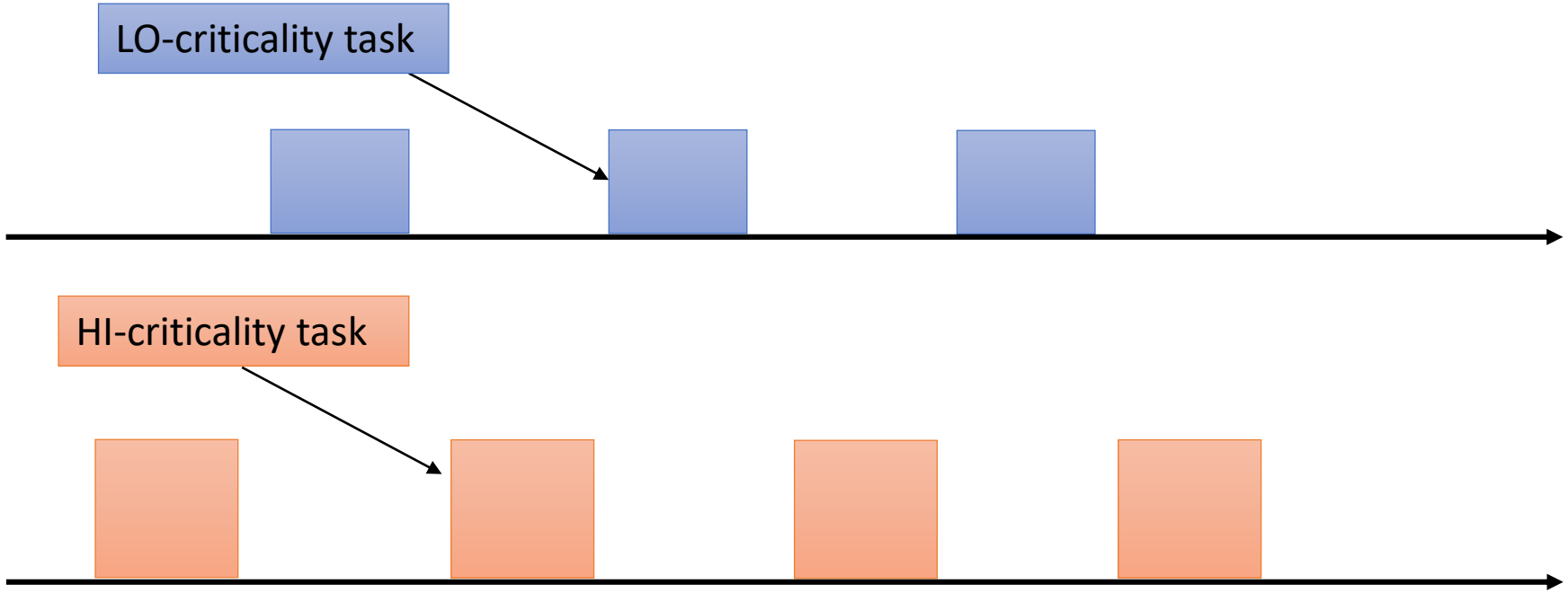
[1]University of Central Florida, [2]Missouri University of Science & Technology, [3]Texas Tech University,

[*]Alphabetically ordered by last names

RTNS'19, Nov. 06~08, 2019, Toulouse, France

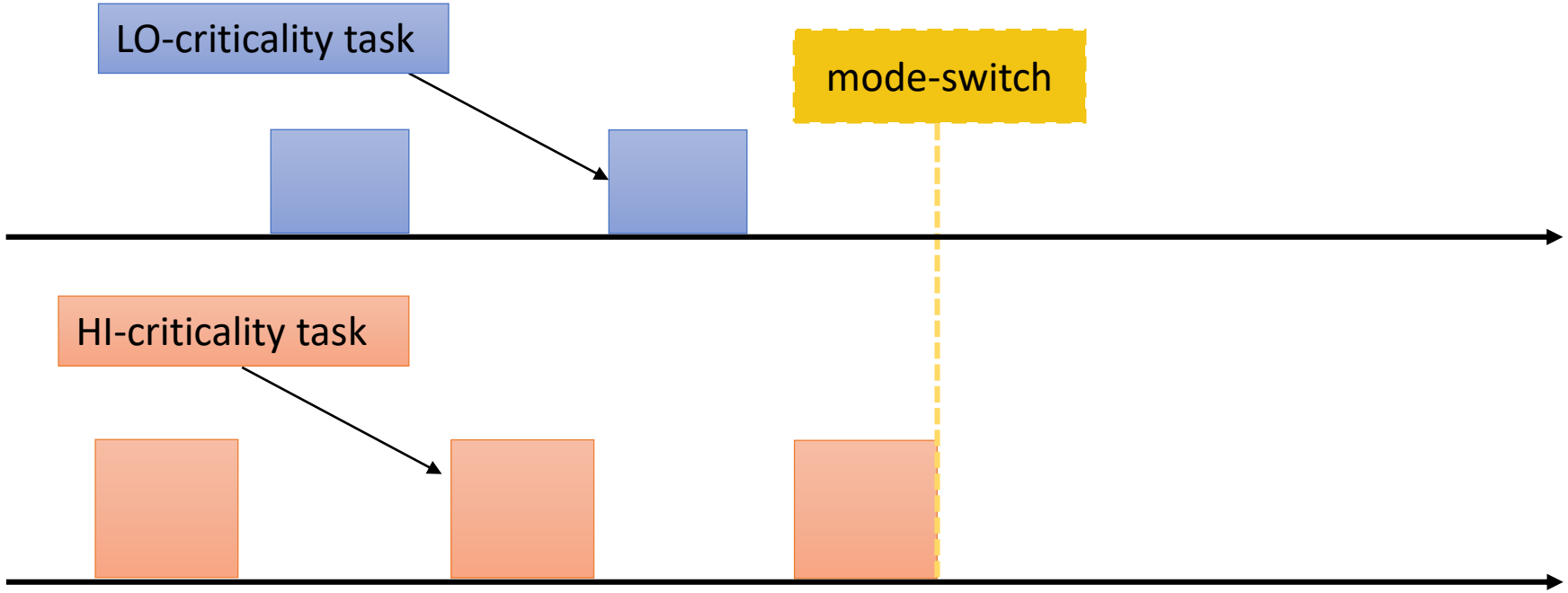
Problem and Motivation

- Traditional MC task model



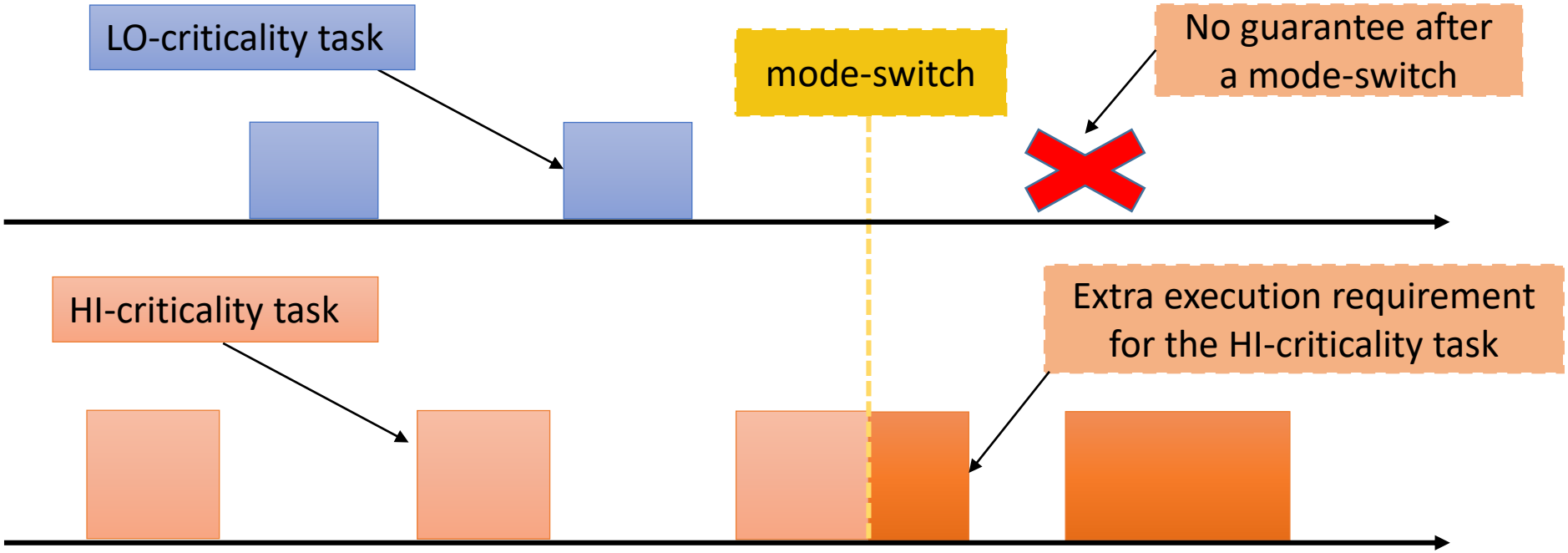
Problem and Motivation

- Traditional MC task model



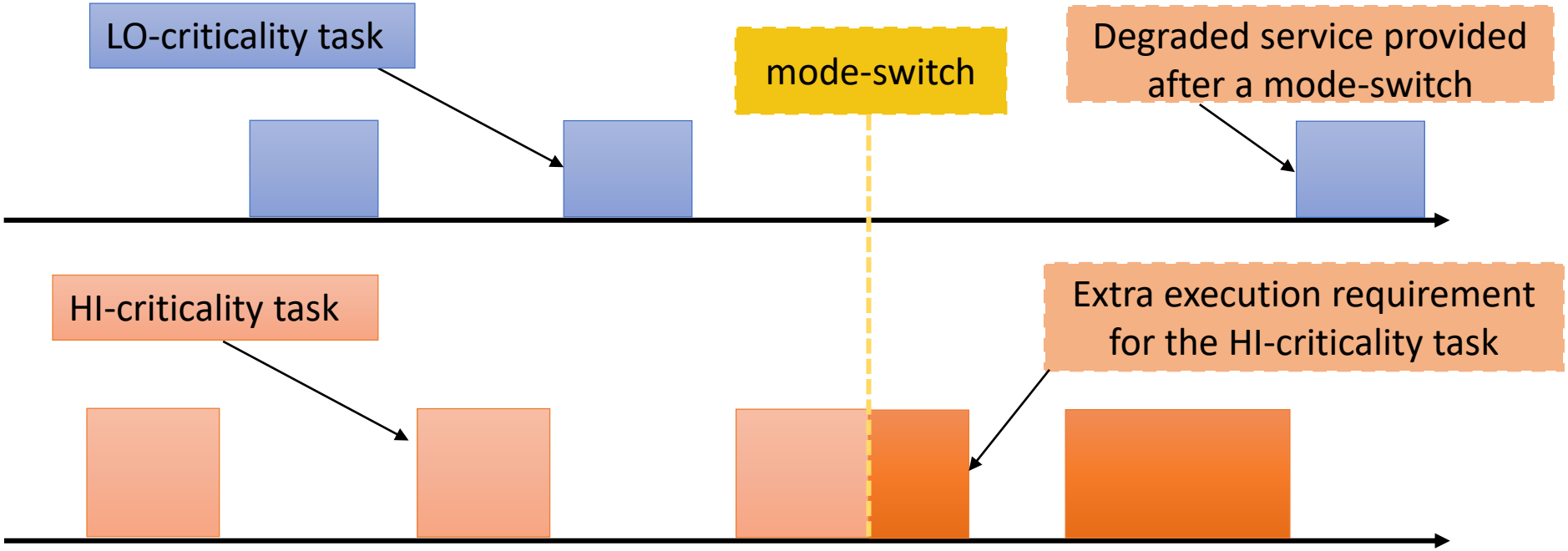
Problem and Motivation

- Traditional MC task model



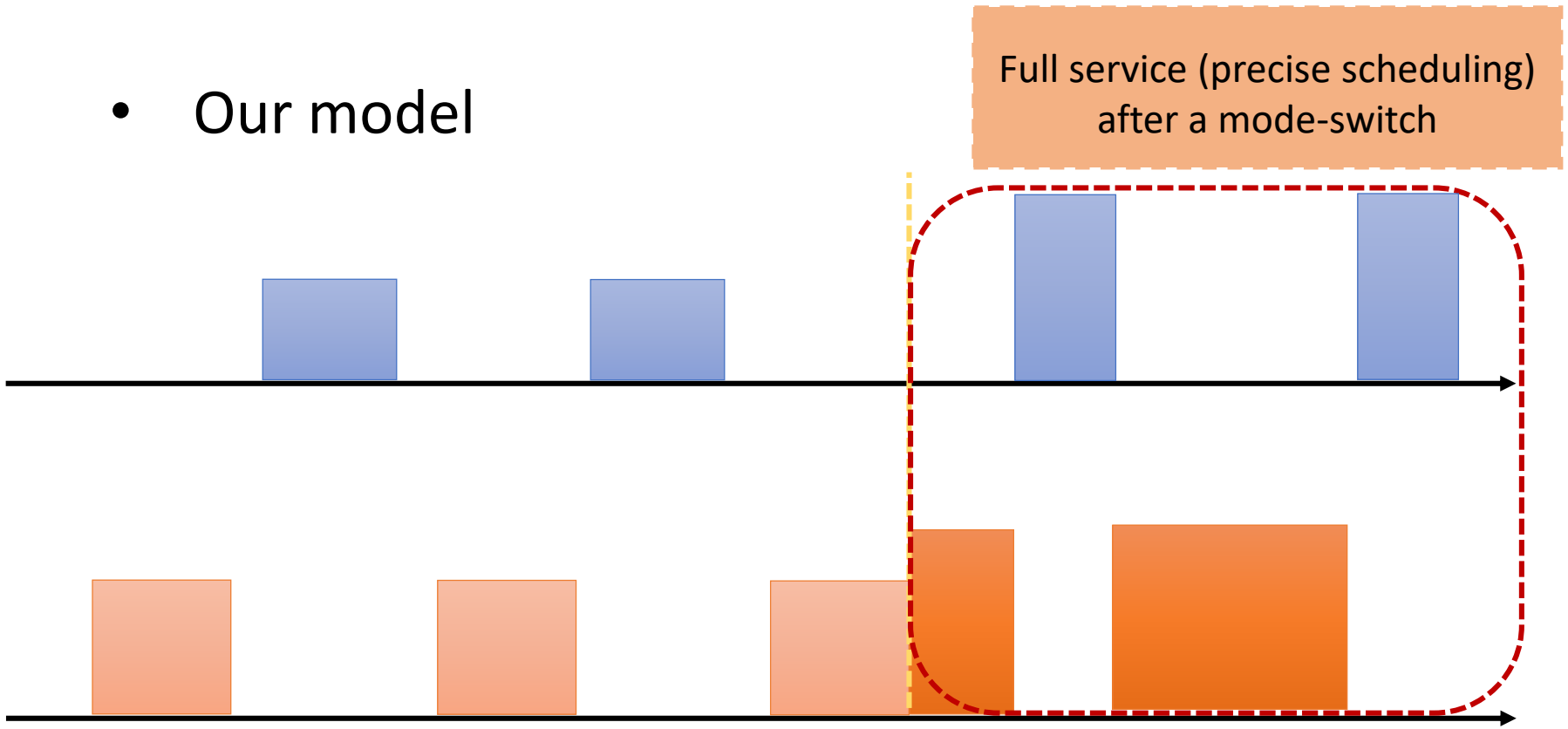
Problem and Motivation

- Traditional MC task model



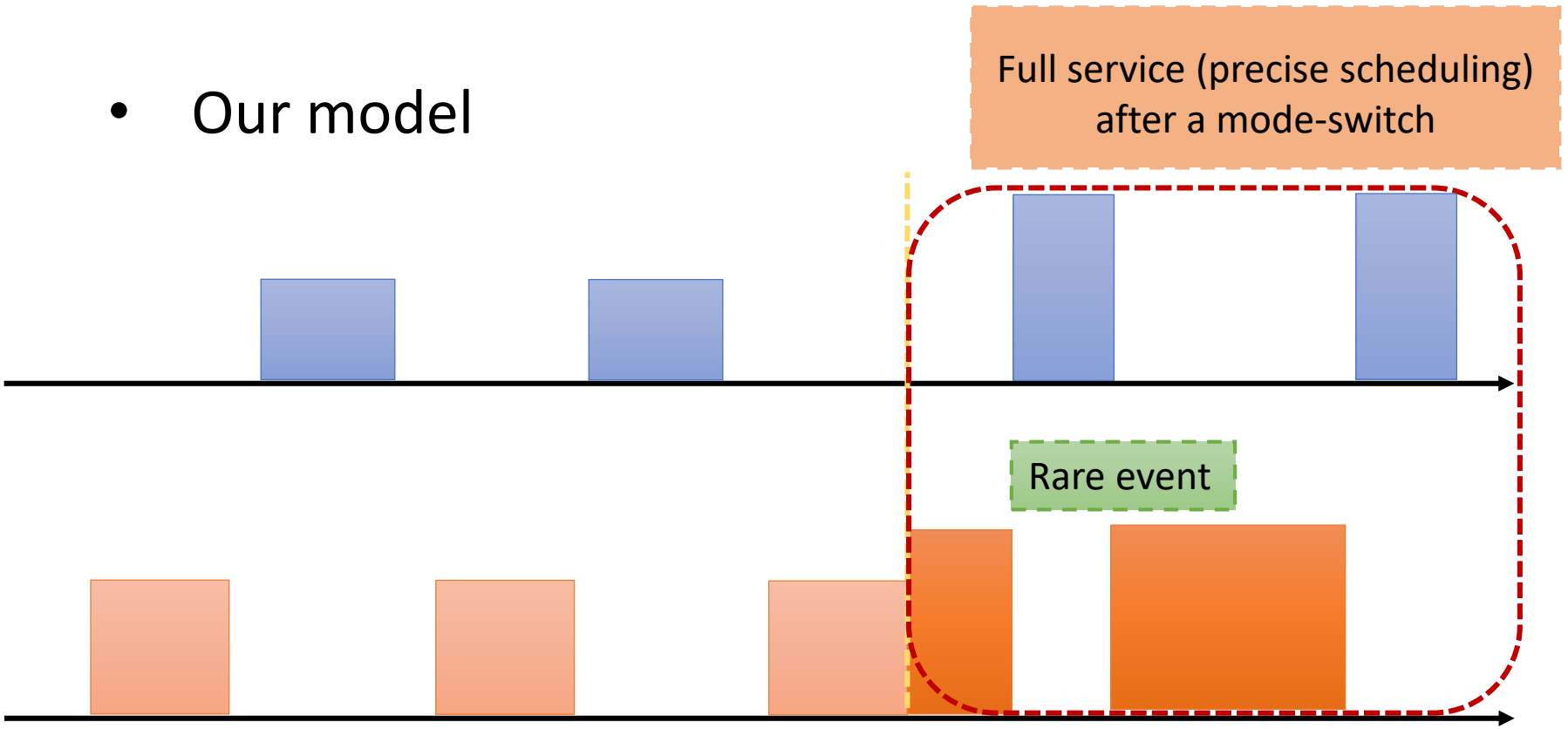
Problem and Motivation

- Our model



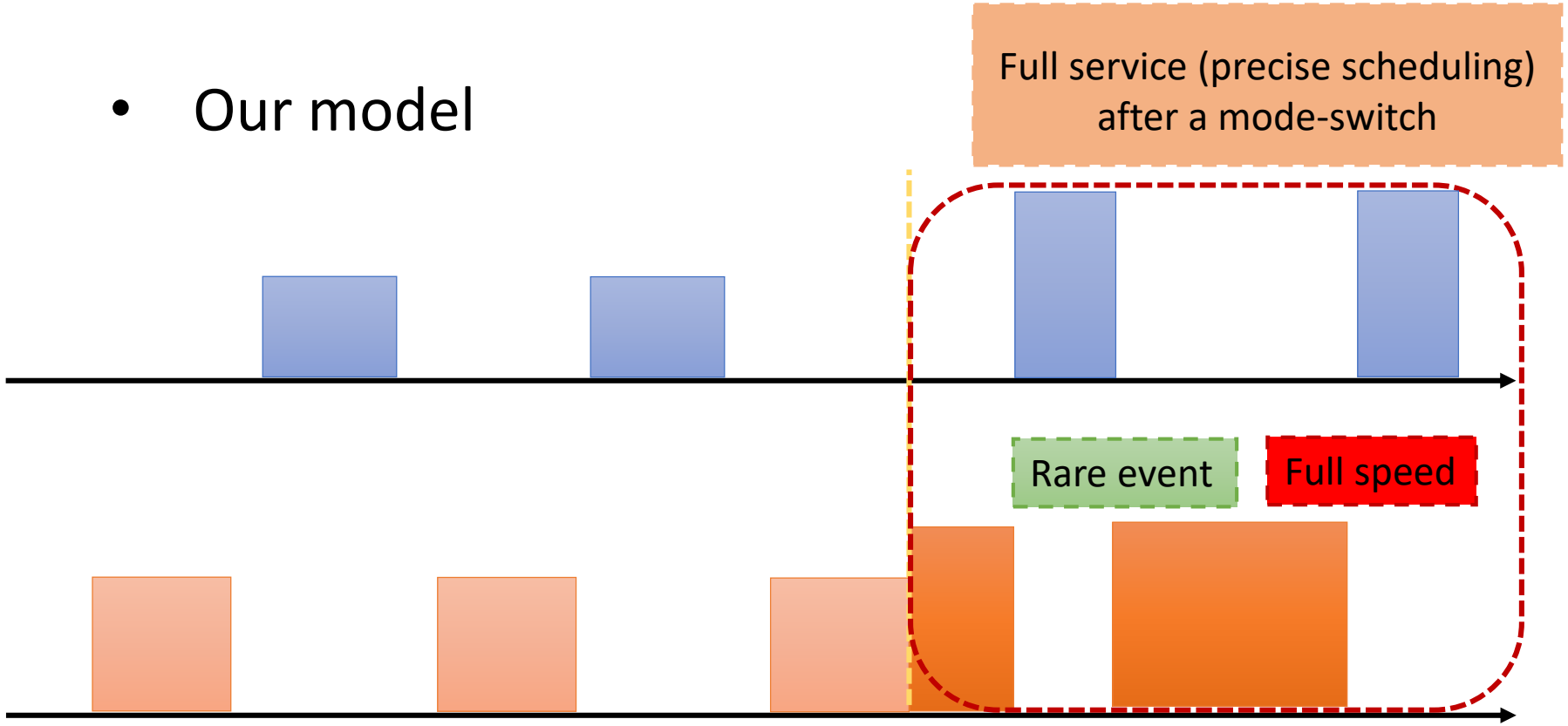
Problem and Motivation

- Our model



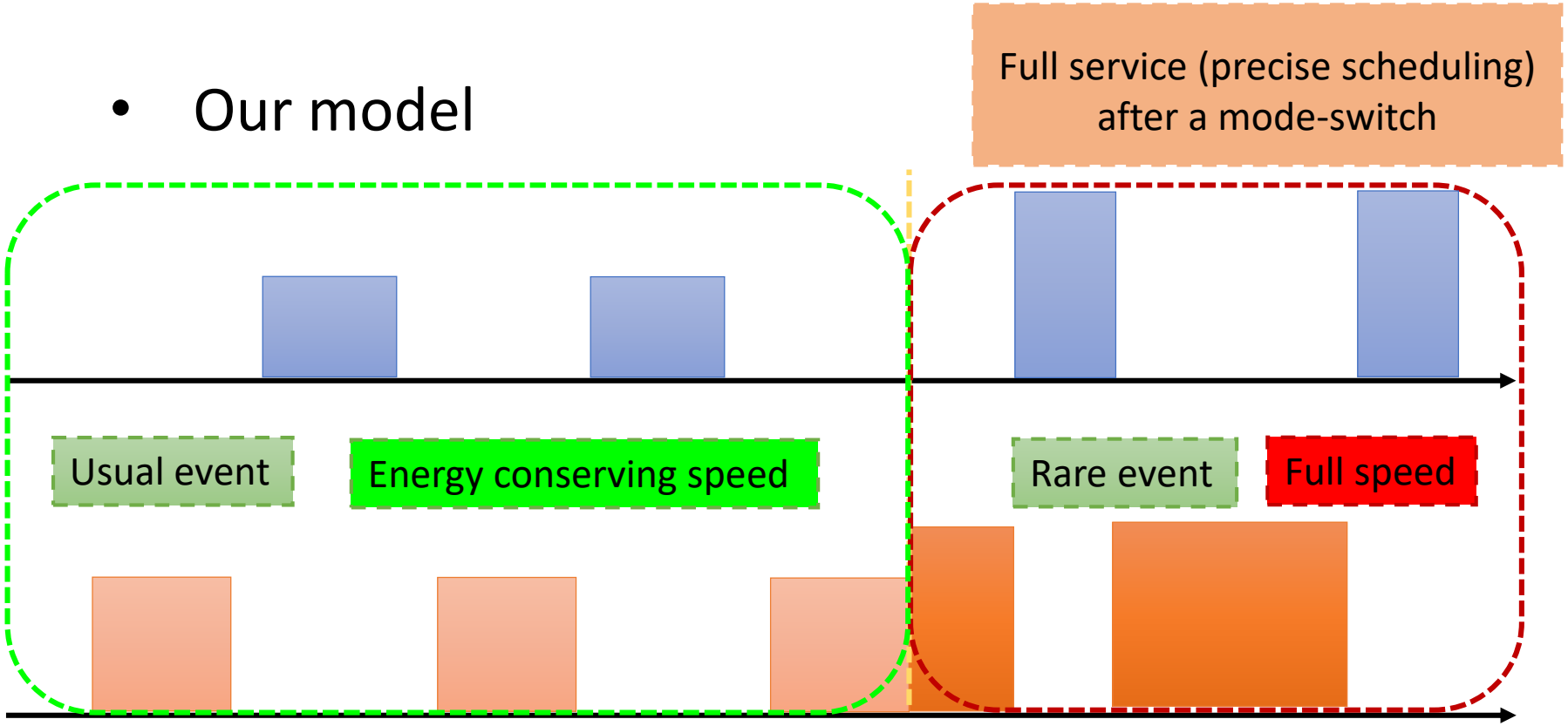
Problem and Motivation

- Our model



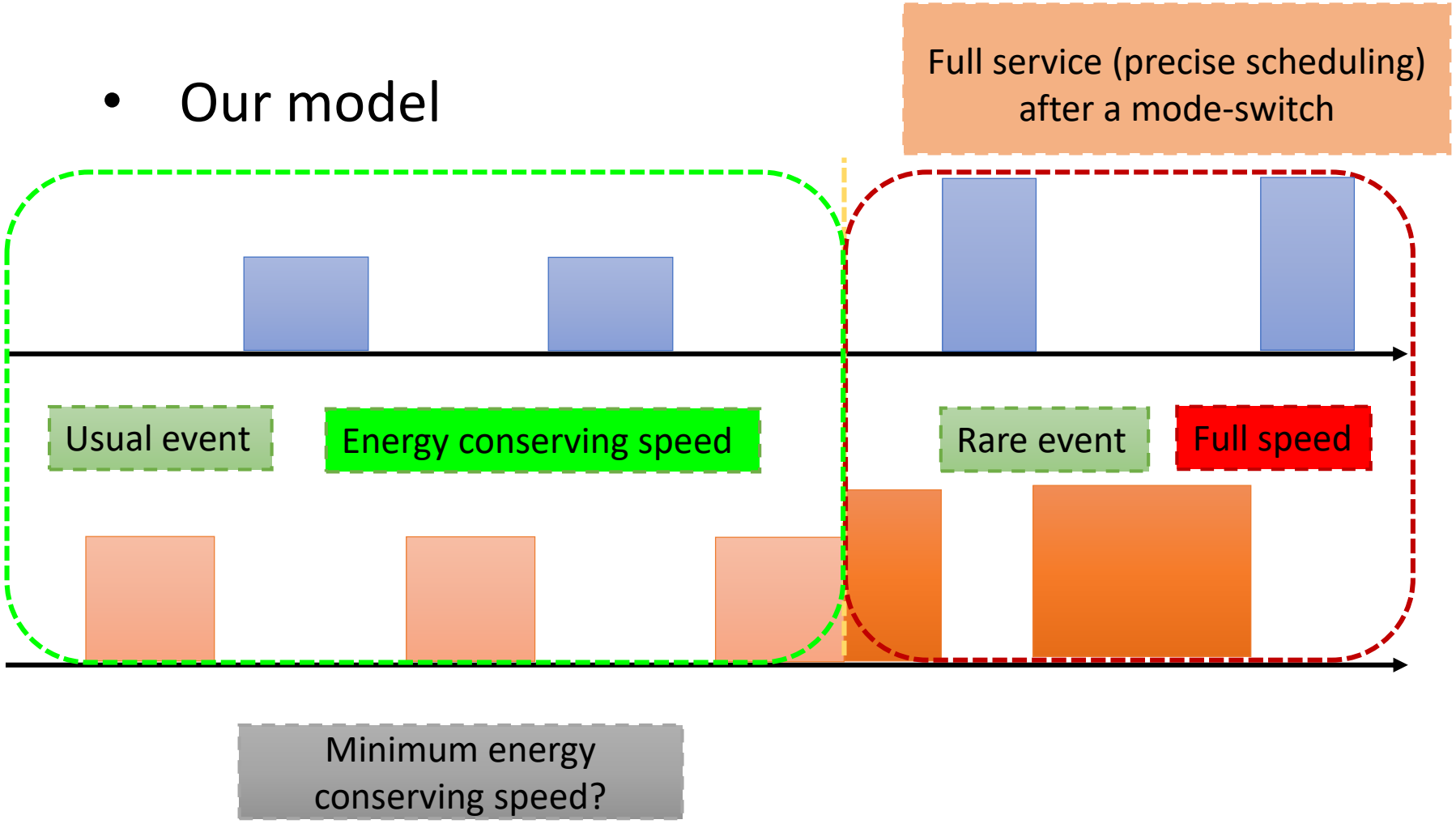
Problem and Motivation

- Our model



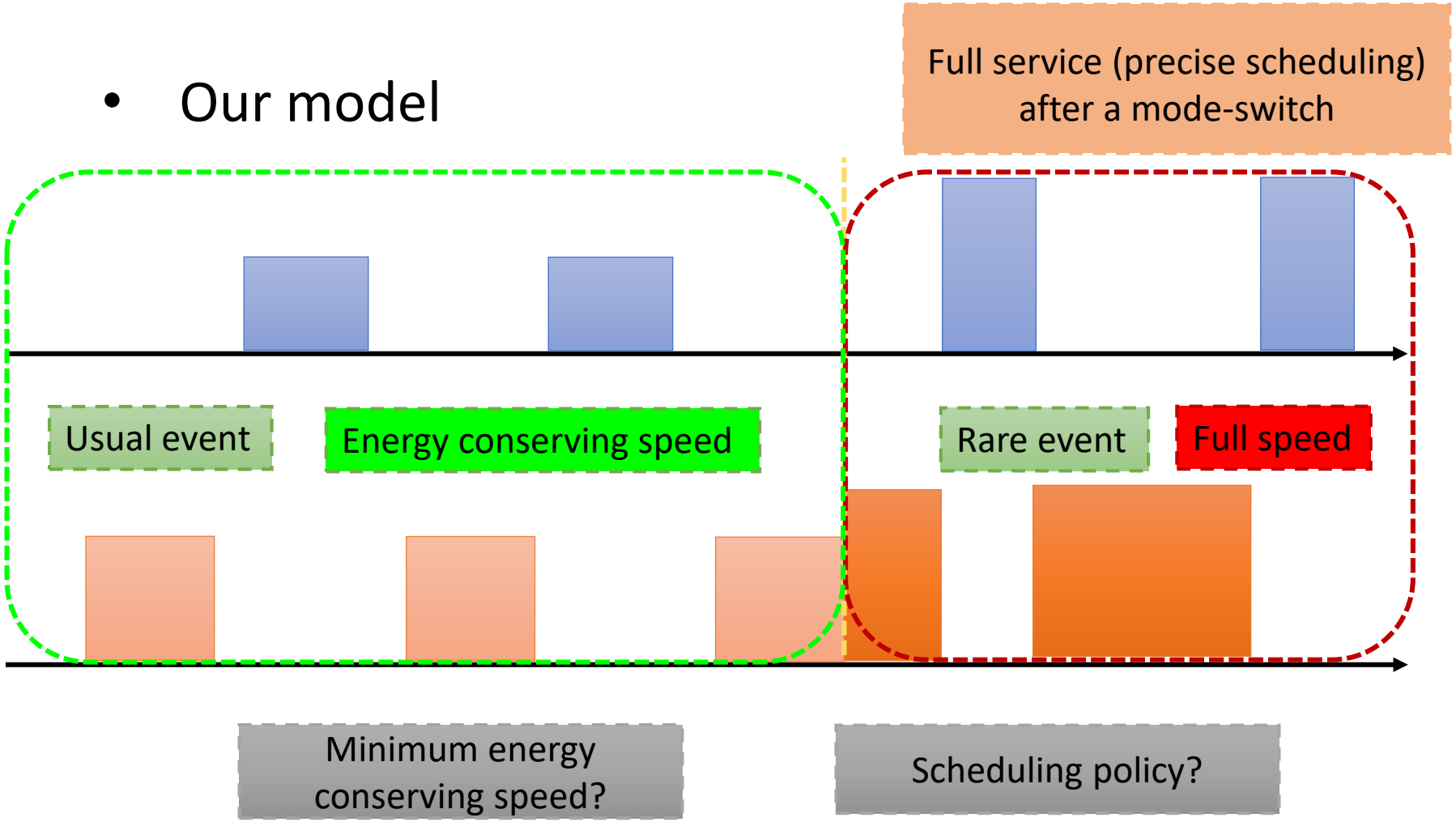
Problem and Motivation

- Our model



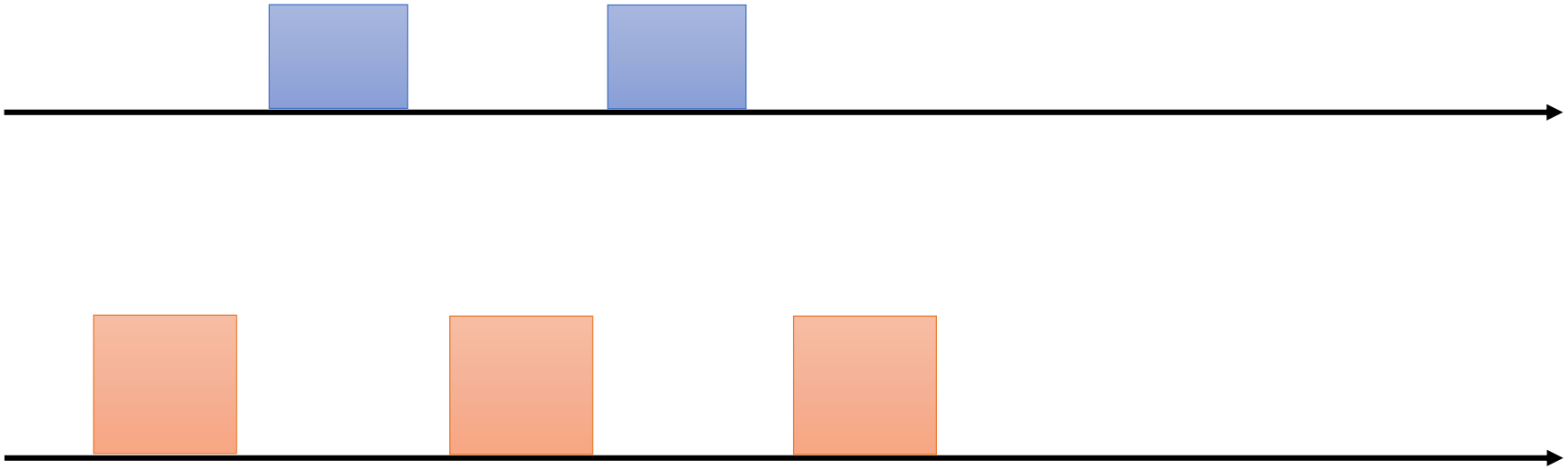
Problem and Motivation

- Our model



EDF-VD

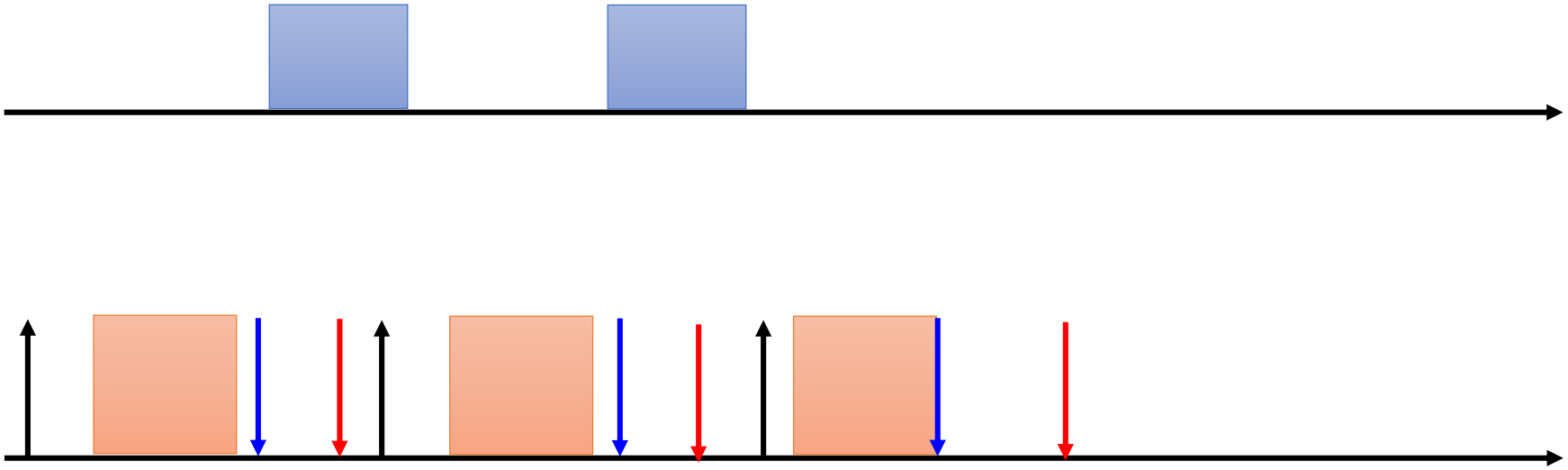
- Setting a virtual deadline (**VD**) to all the HI-criticality tasks at LO-criticality mode.



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

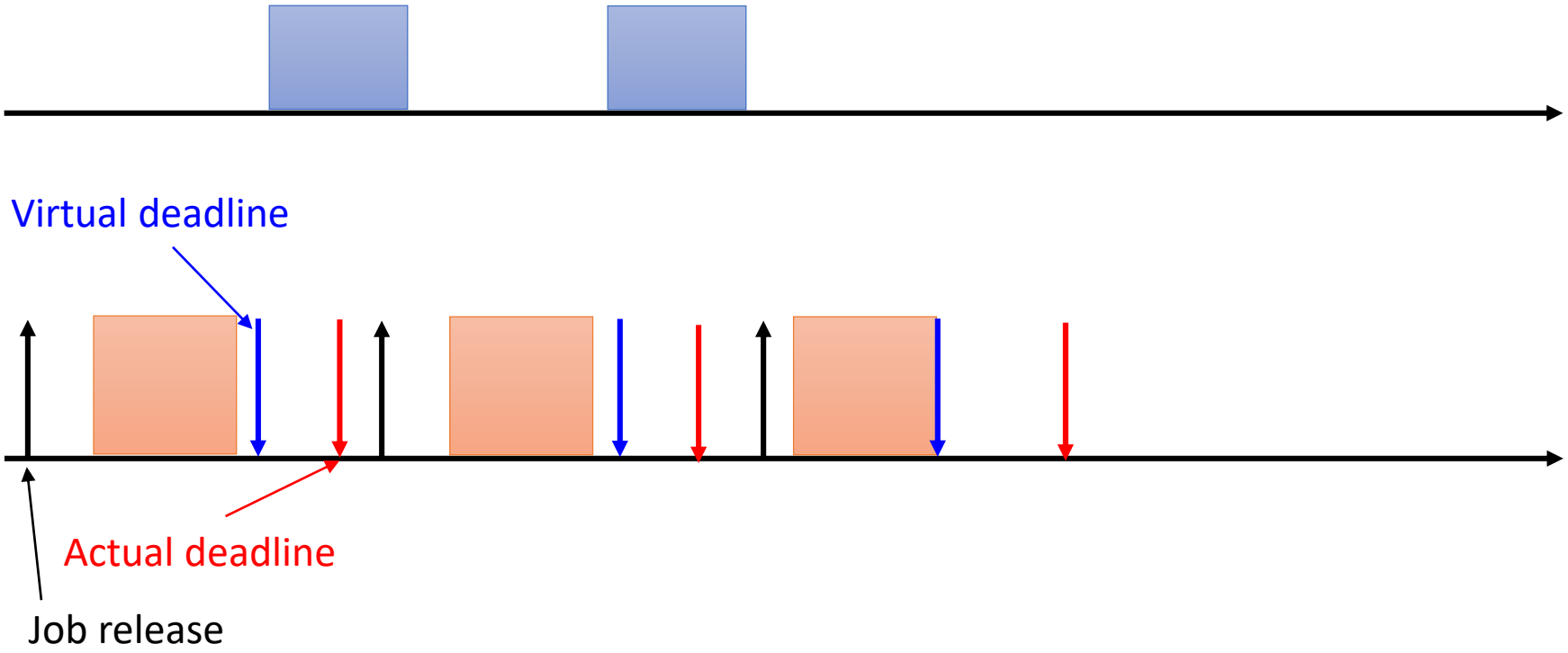
EDF-VD

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



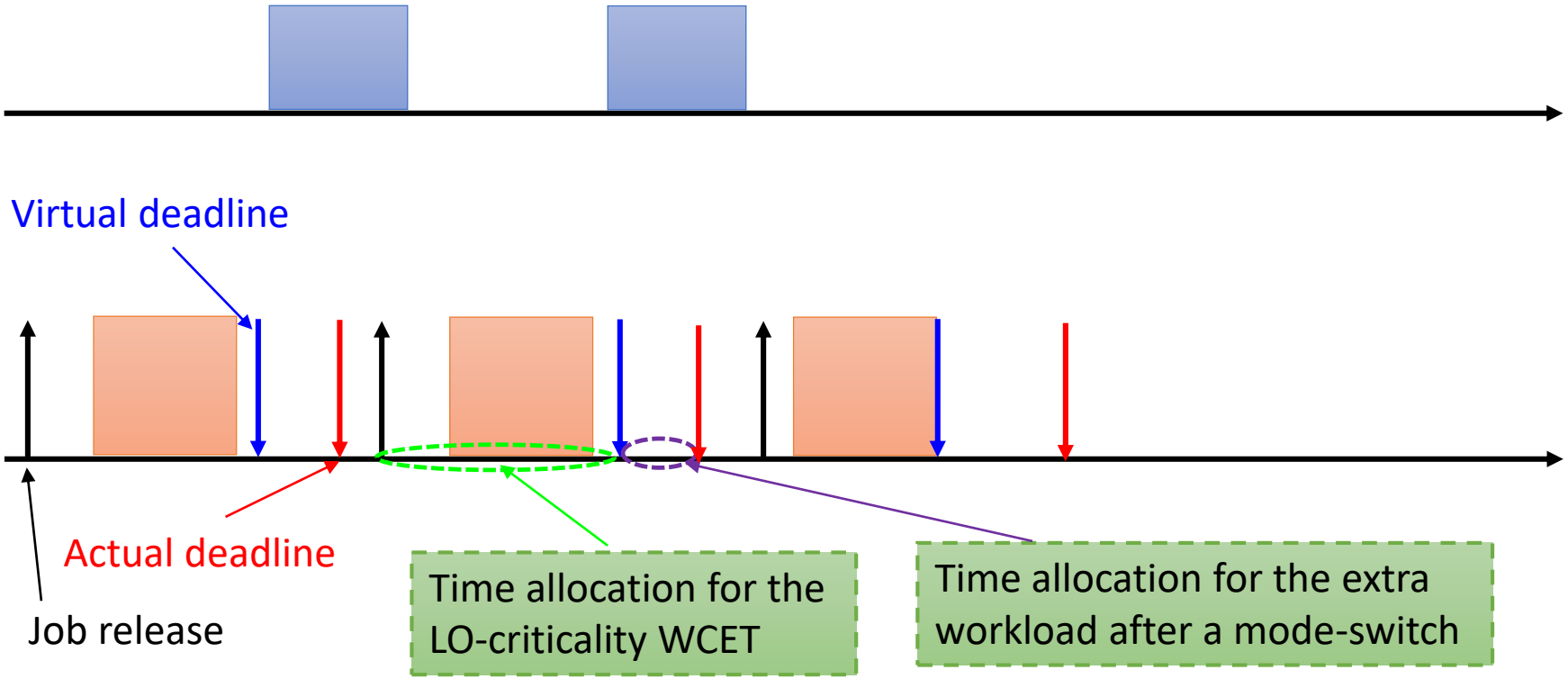
EDF-VD

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



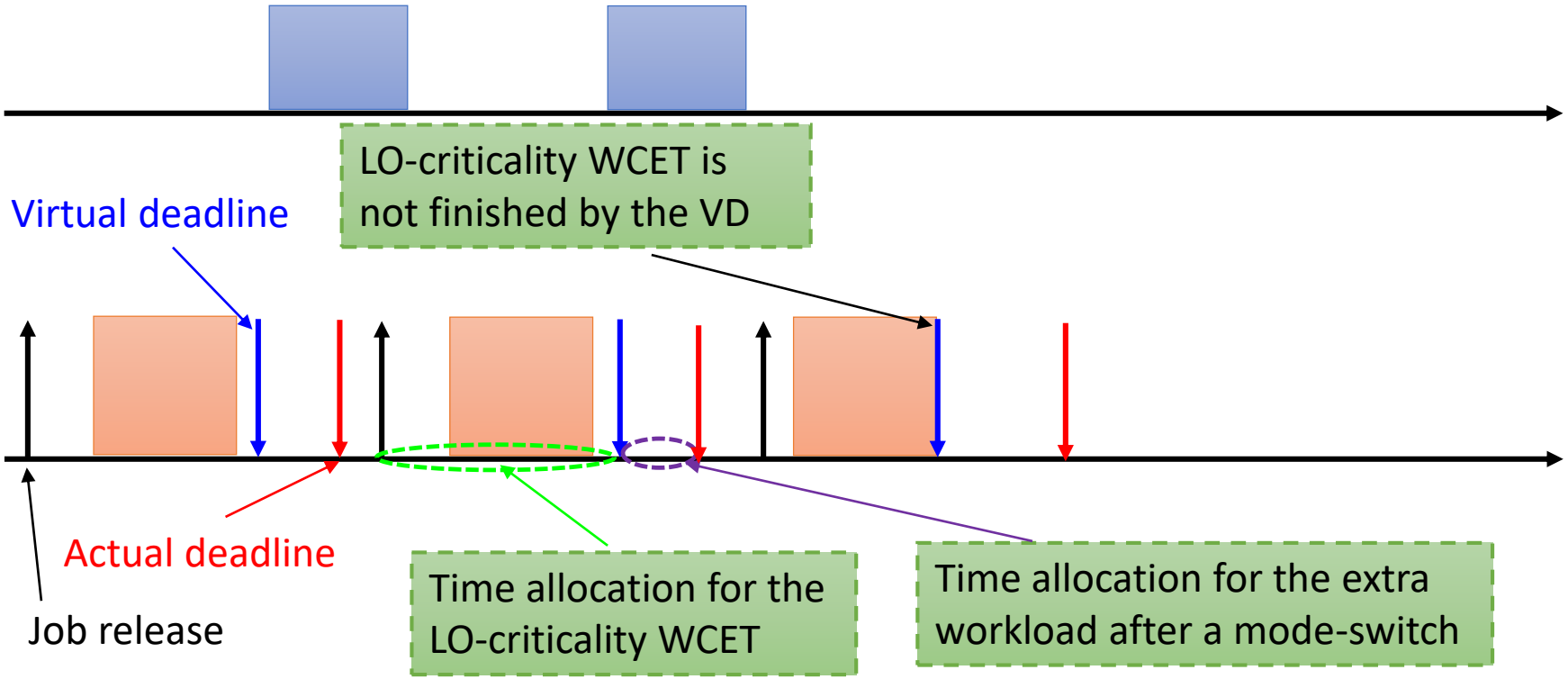
EDF-VD

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



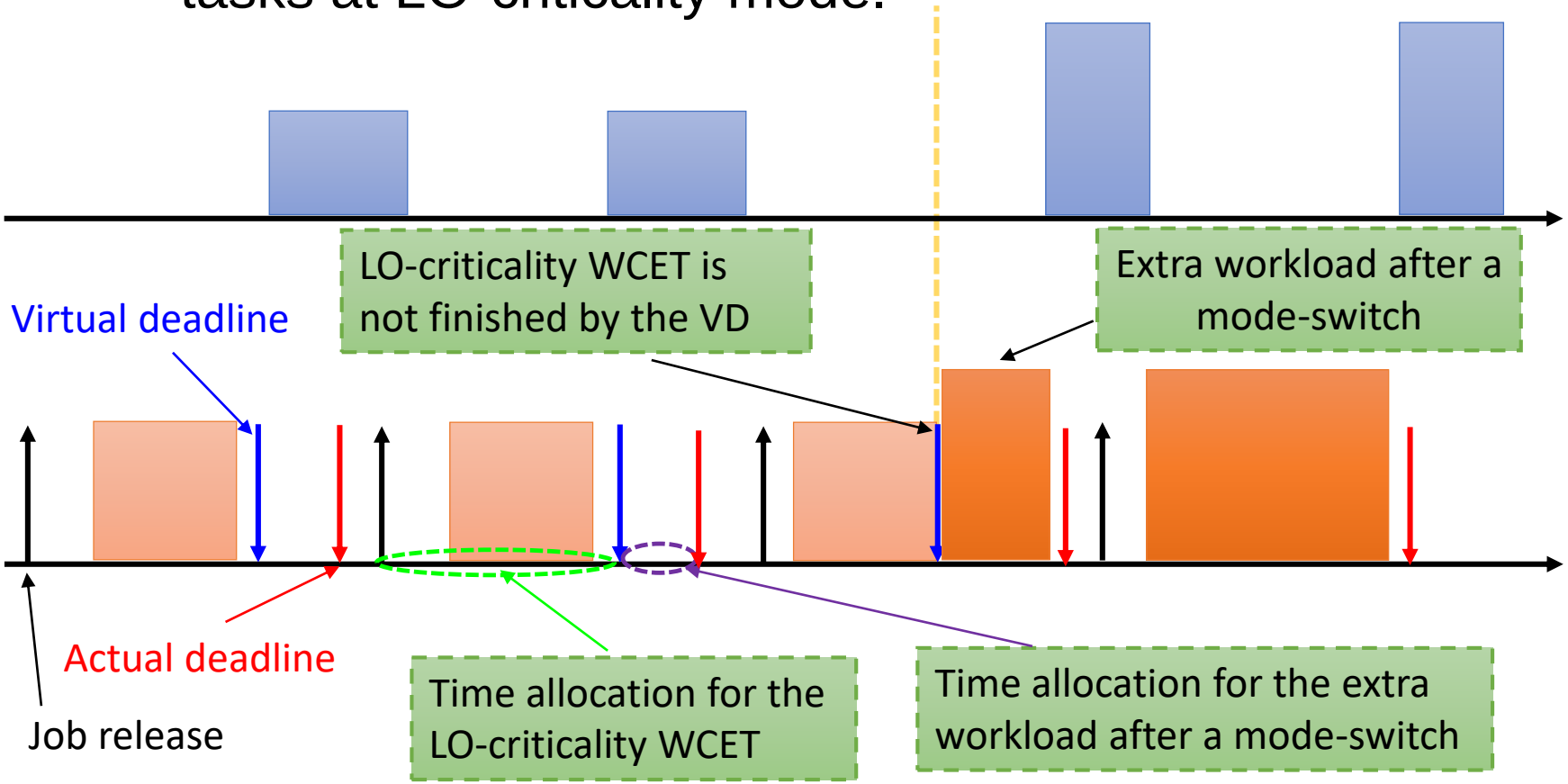
EDF-VD

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



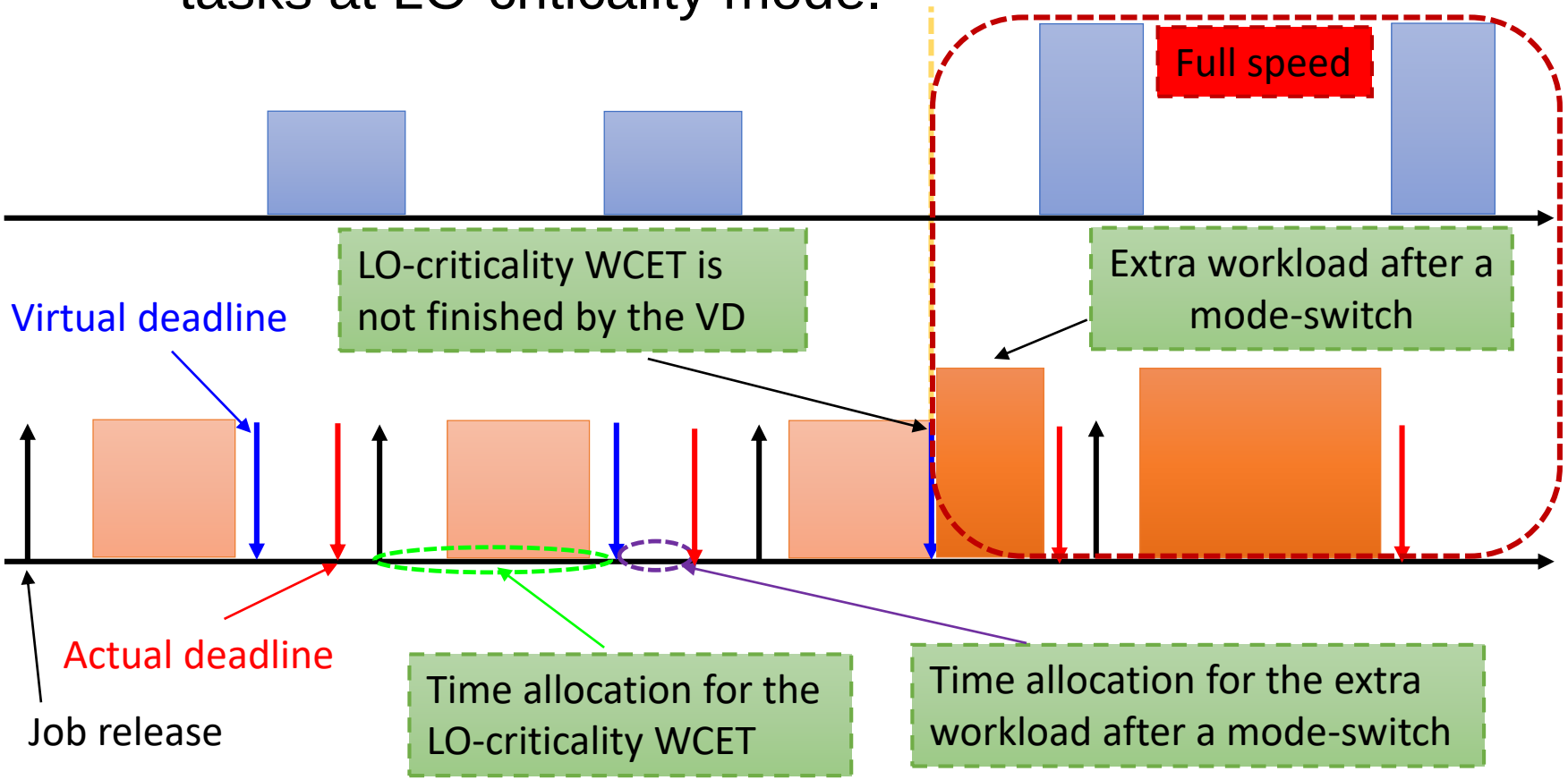
EDF-VD

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



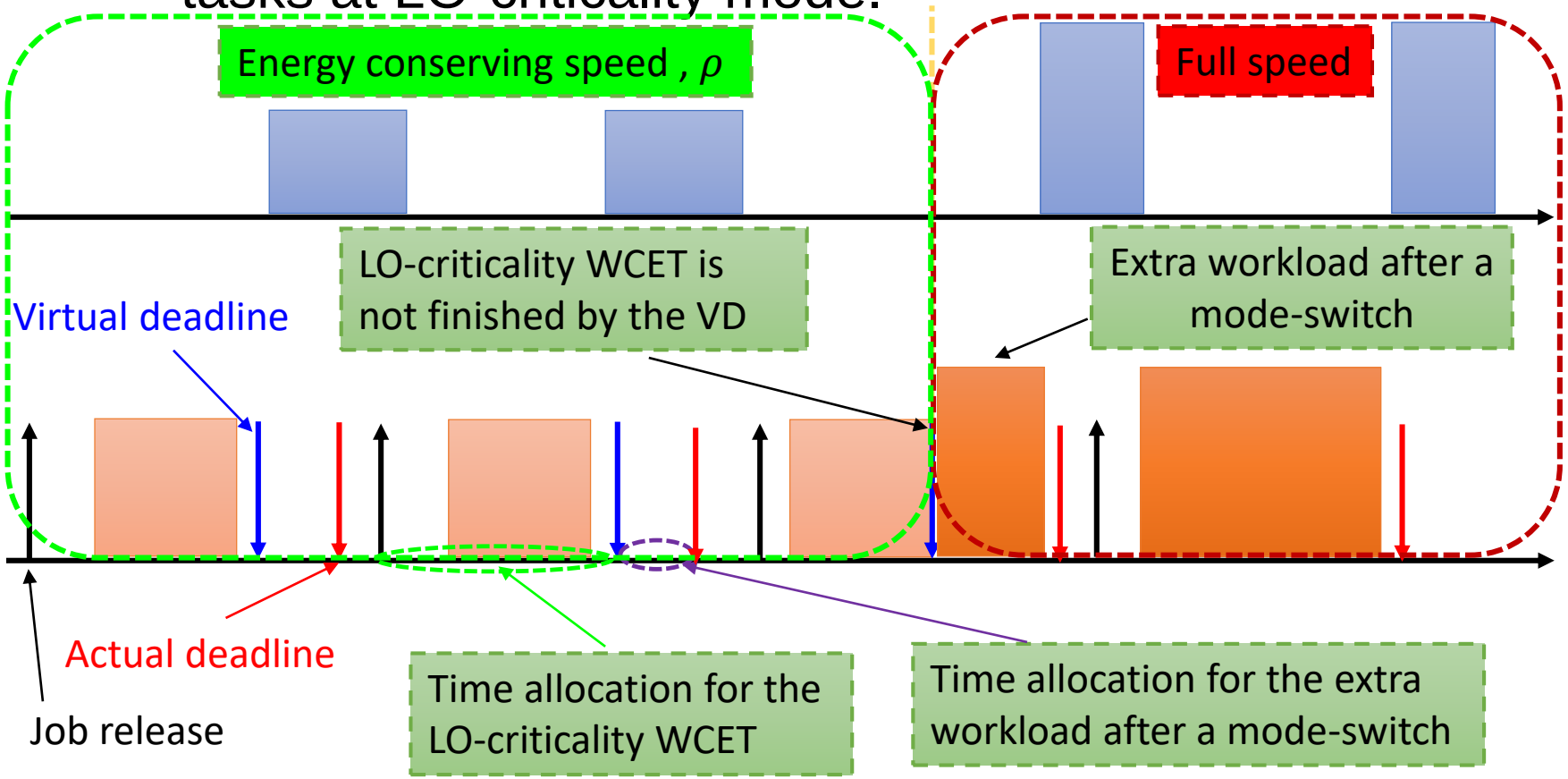
EDF-VD

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



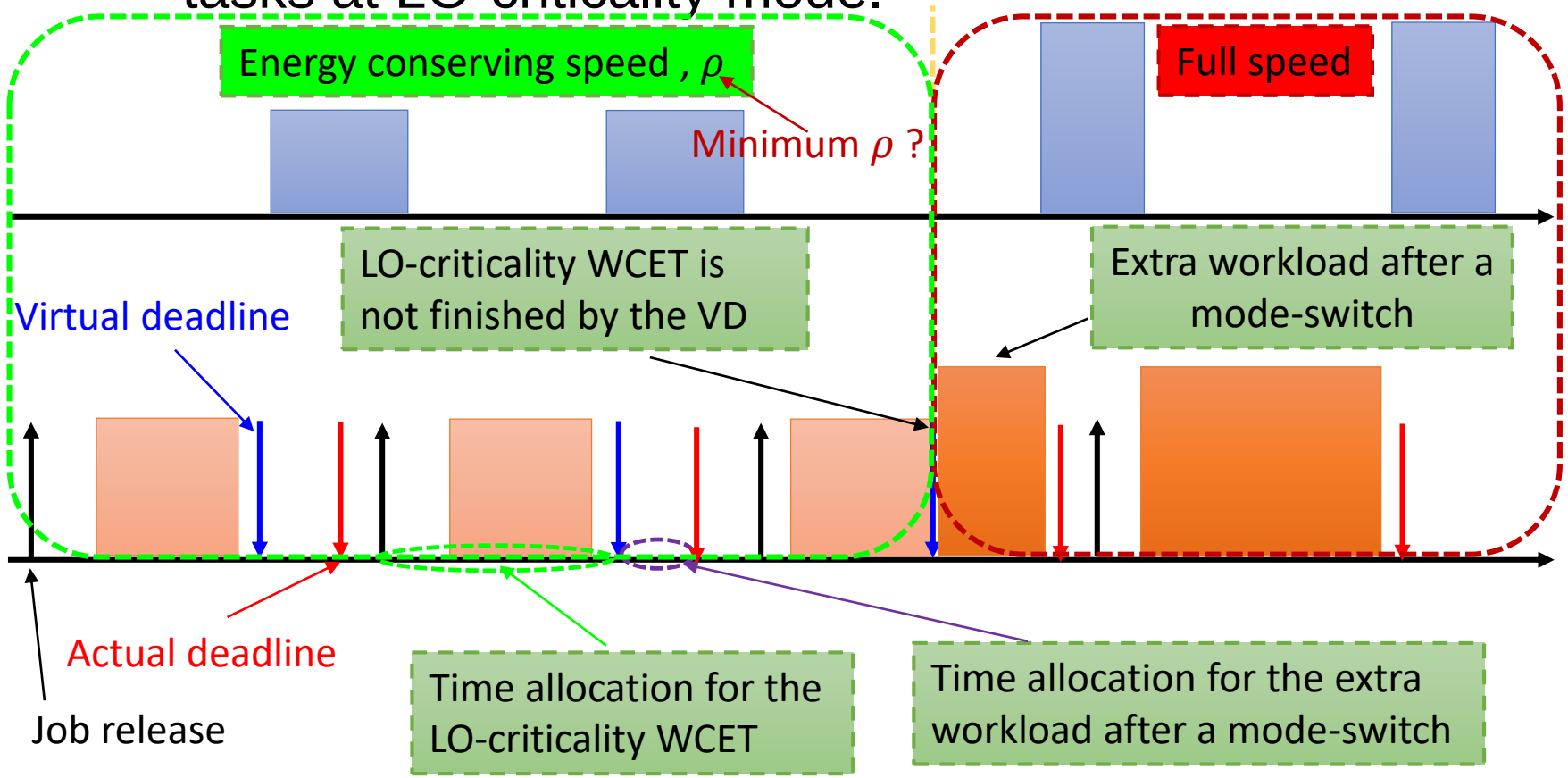
EDF-VD

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



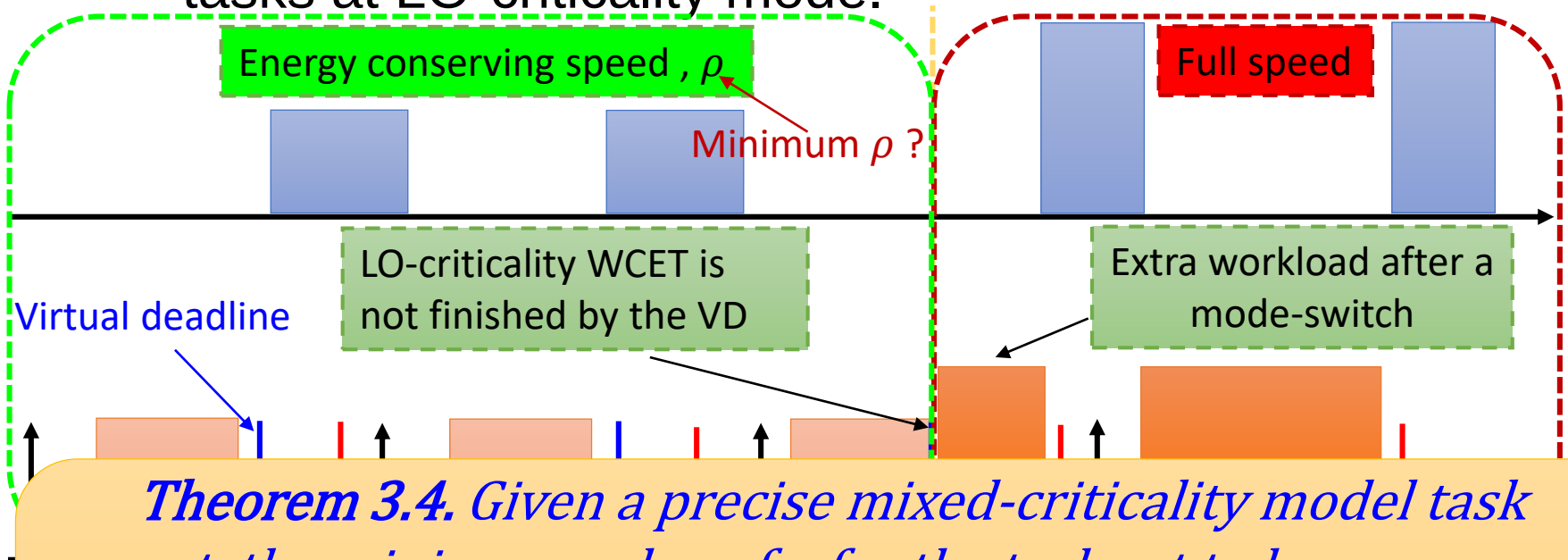
EDF-VD

- 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.

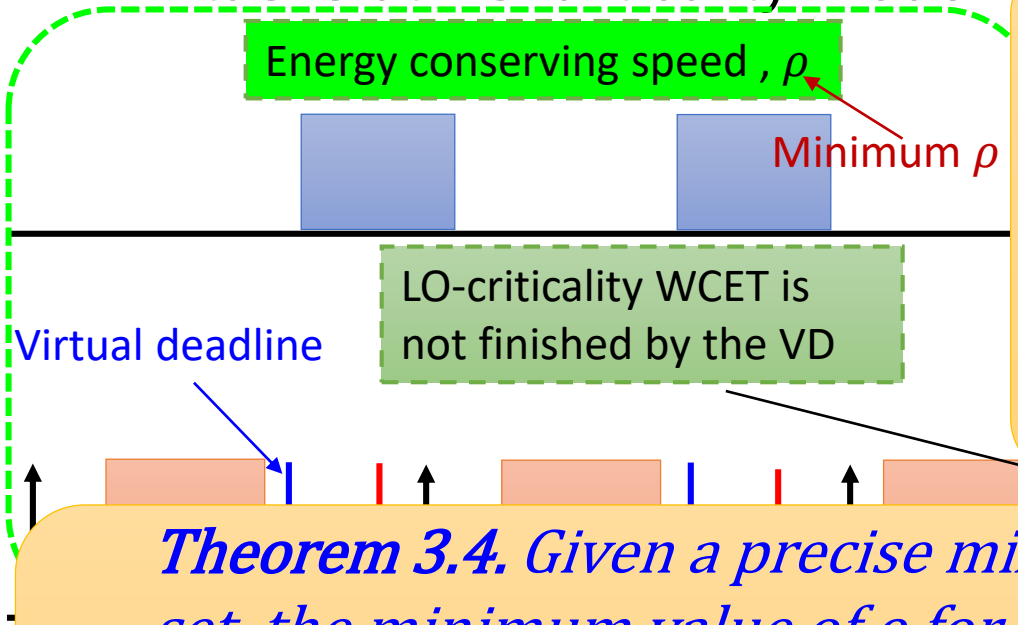


Theorem 3.4. *Given a precise mixed-criticality model task set, the minimum value of ρ for the task set to be schedulable by EDF-VD is*

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

EDF-VD

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



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

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

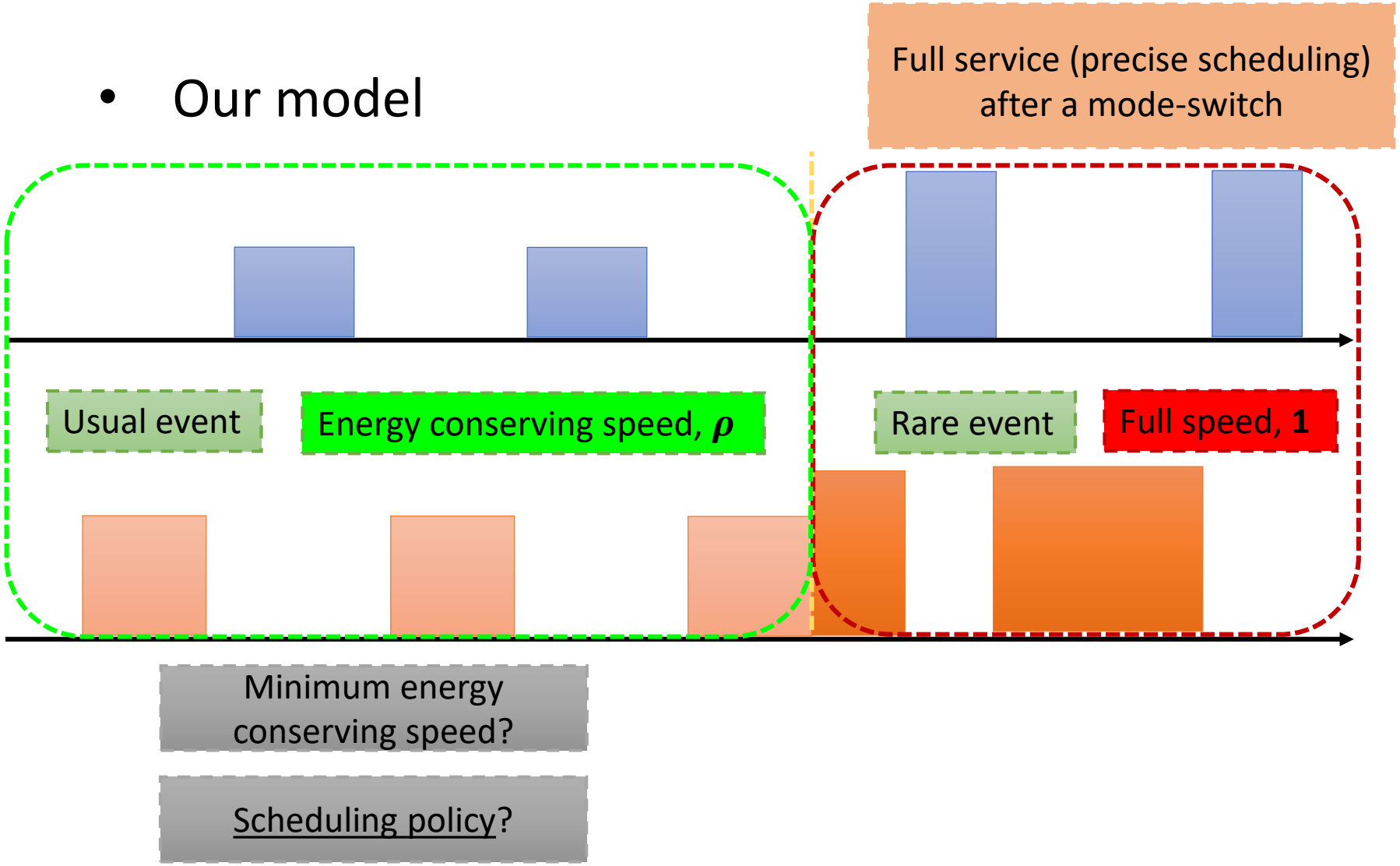
$$x \ge \frac{U_H{}^L}{(\rho - U_L^L)}$$

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*

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

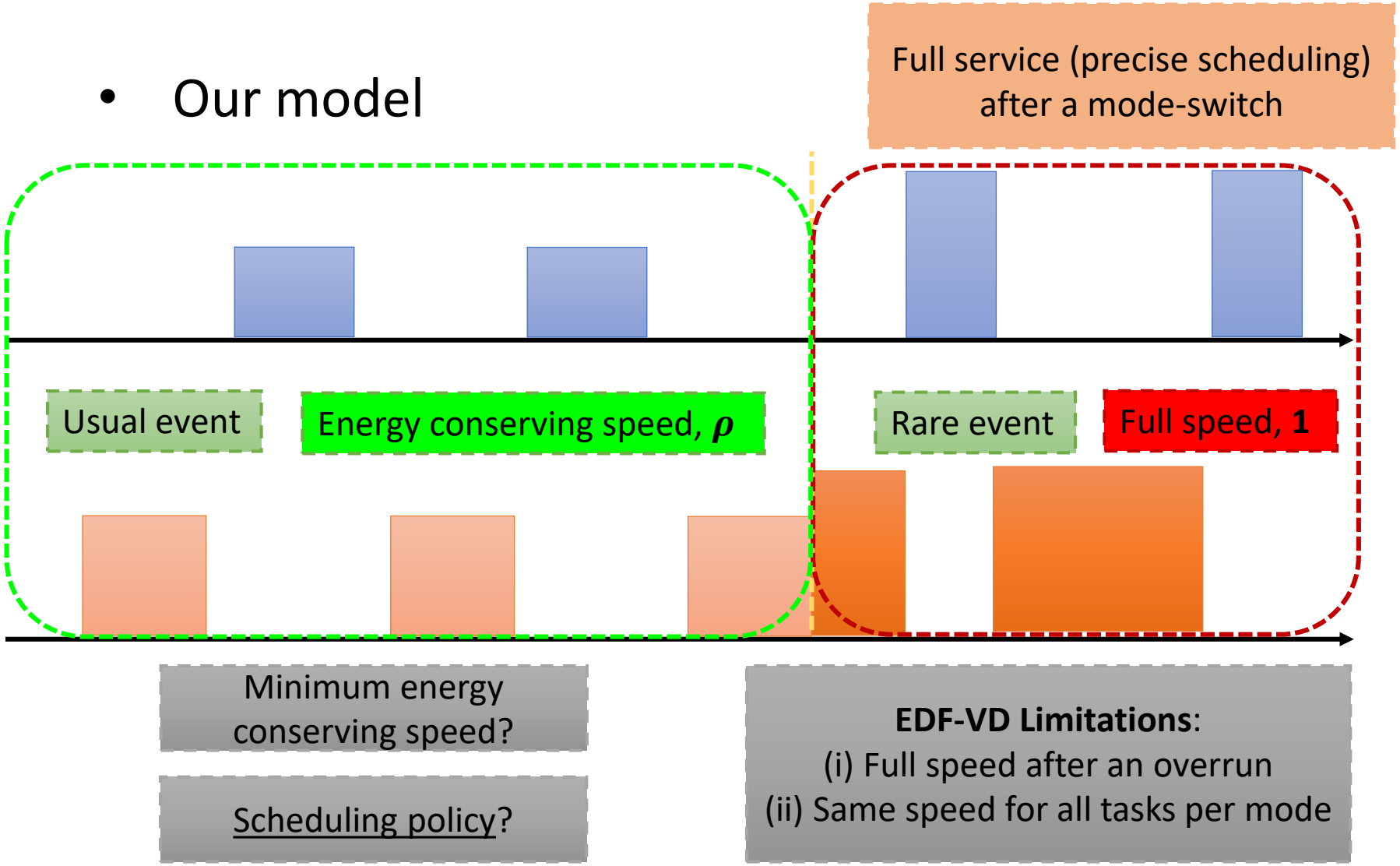
Problem and Motivation

- Our model



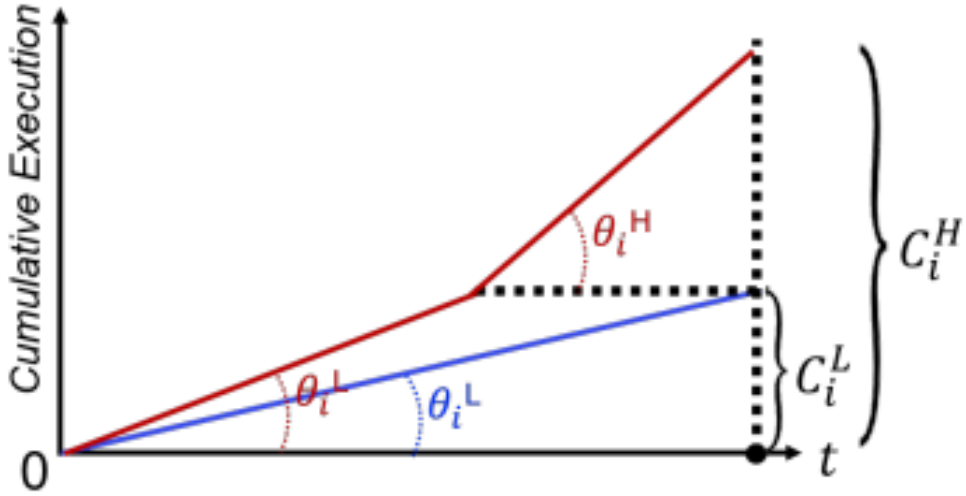
Problem and Motivation

- Our model



MC-Fluid (MCF)

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

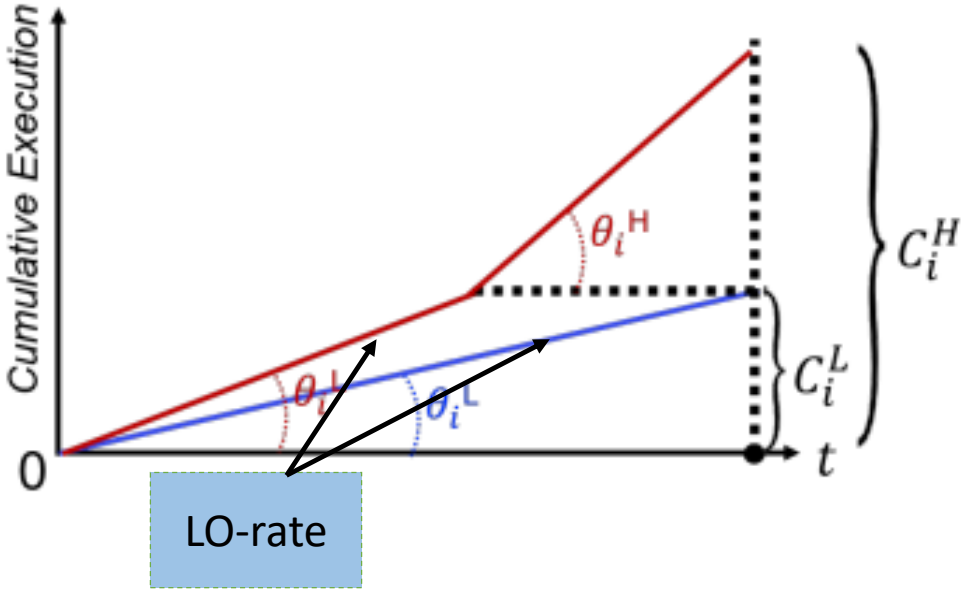


Acknowledgment: Sanjoy Baruah

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*.

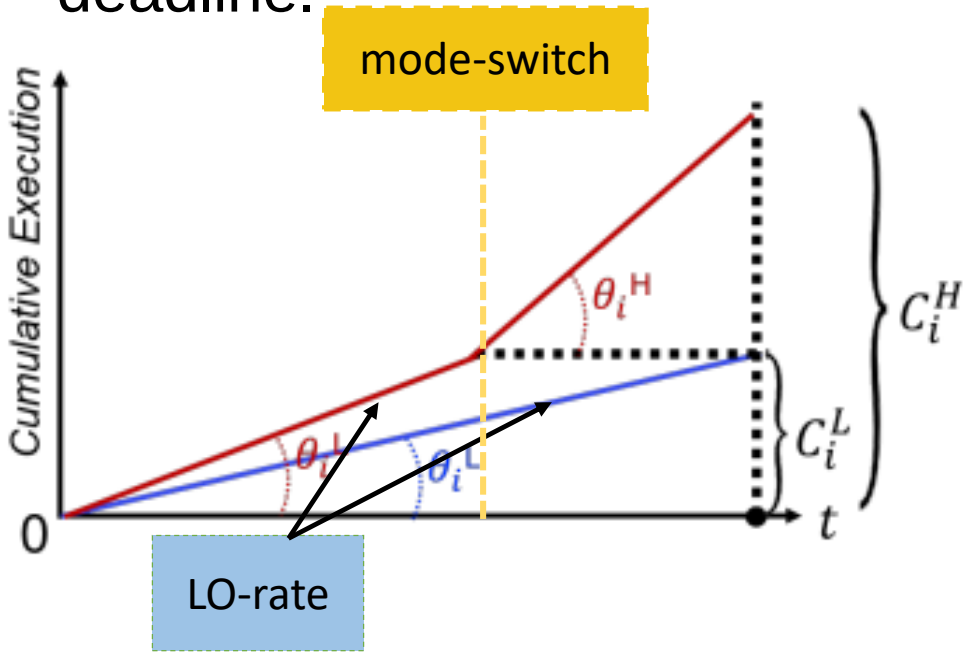
MC-Fluid (MCF)

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



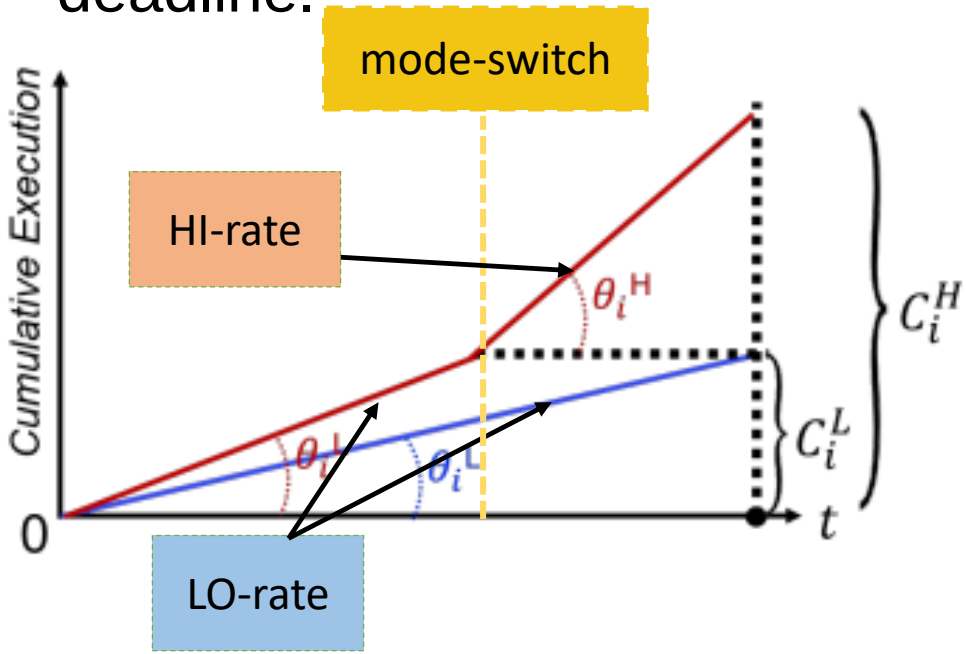
MC-Fluid (MCF)

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



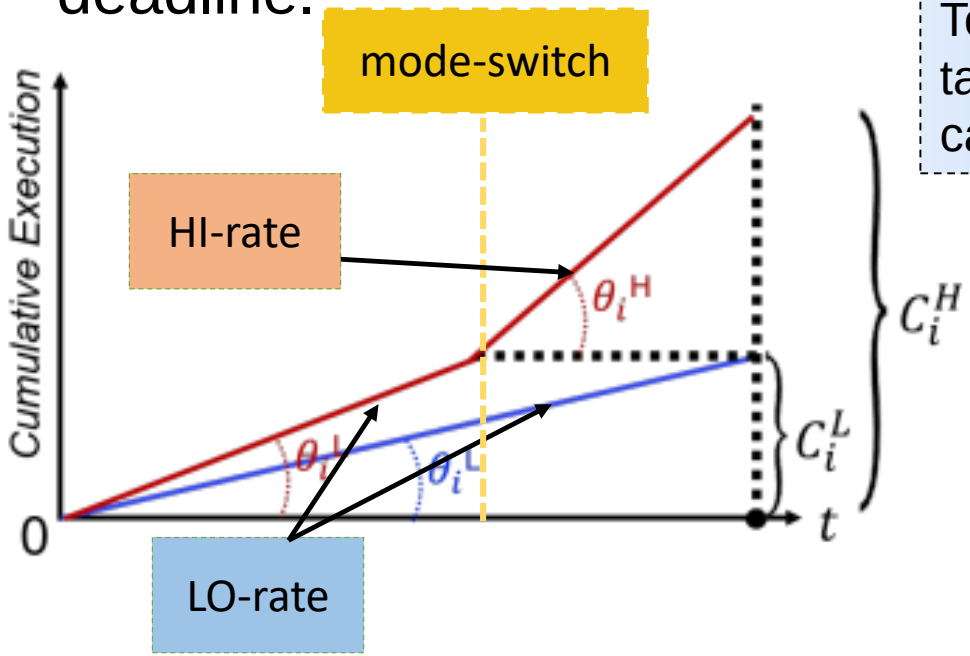
MC-Fluid (MCF)

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



MC-Fluid (MCF)

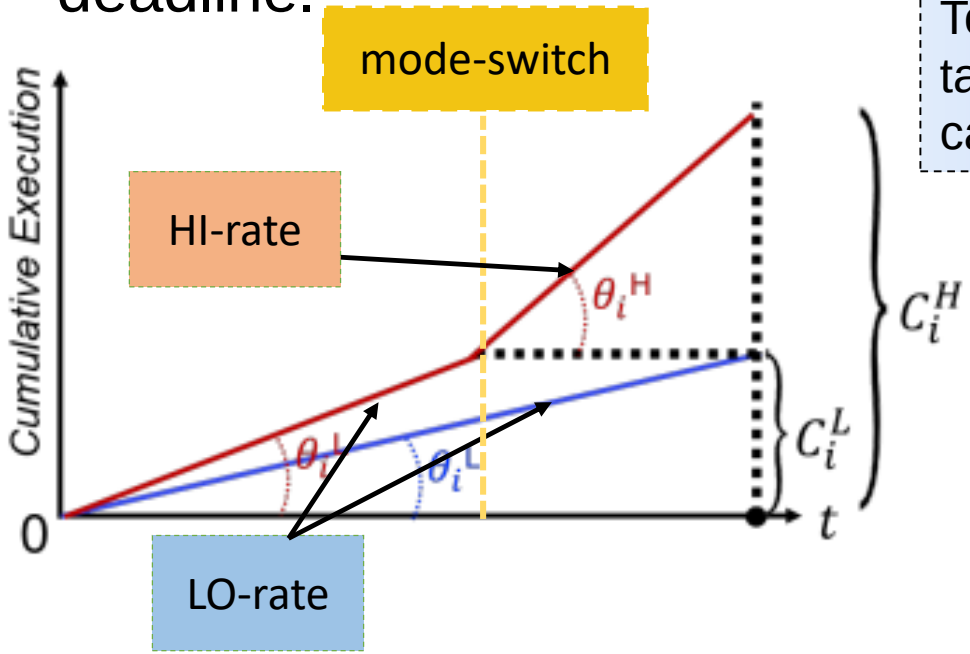
- All the tasks receive processor-share and have a constant execution rate from their release to the deadline.



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

MC-Fluid (MCF)

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



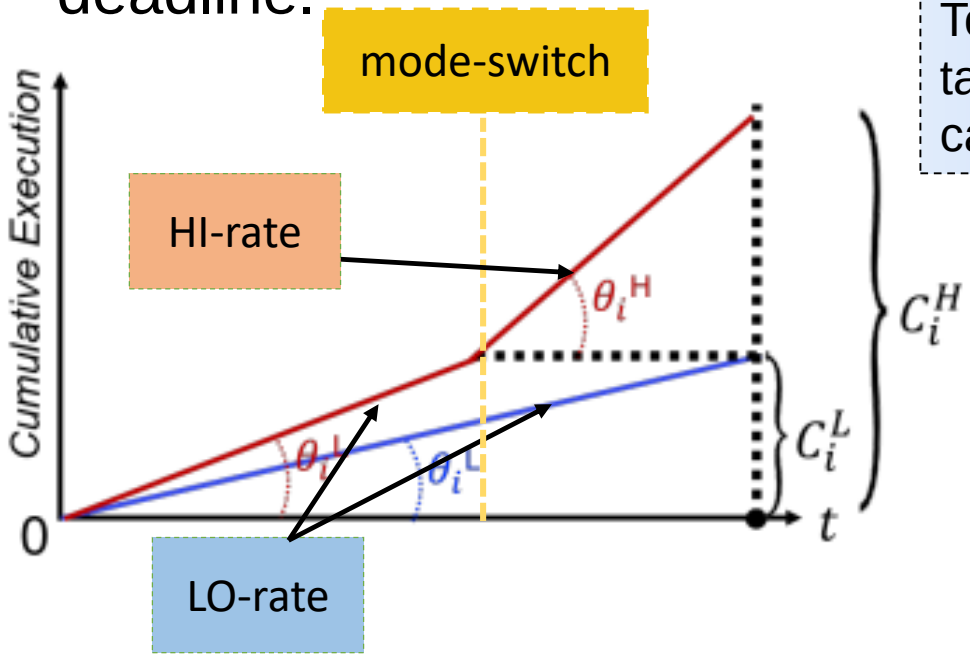
Total processor-share (of each task) is less or equal to the capacity (speed) of the processor.

$$\sum_{\tau_i \in \tau} \theta_i^l < 1$$

$$\sum_{\tau_i \in \tau_H} \theta_i^H \leq 1$$

MC-Fluid (MCF)

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



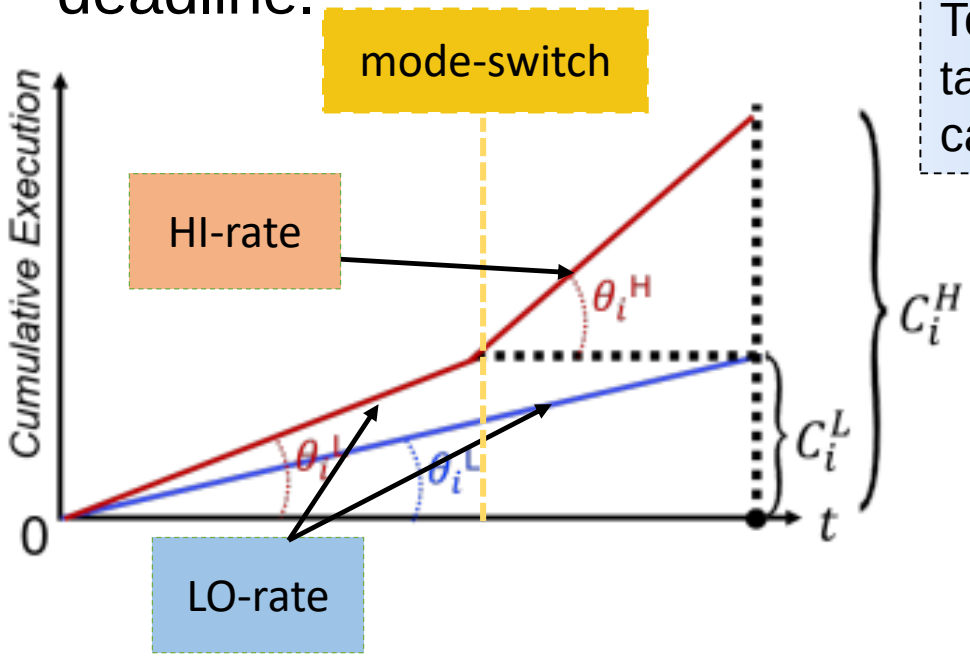
Total processor-share (of each task) is less or equal to the capacity (speed) of the processor.

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

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

MC-Fluid (MCF)

- 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.

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

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

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

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

Performance Evaluation

- **Speedup Bound, s.** How much faster processors are required for an algorithm to schedule the same taskset, scheduled by an optimal algorithm.
- **Approximation Ratio, α.** The ratio of energy conserving speed determined by an algorithm vs the optimal algorithm.

Performance Evaluation

- **Speedup Bound, s.** How much faster processors are required for an algorithm to schedule the same taskset, scheduled by an optimal algorithm.
- **Approximation Ratio, α.** The ratio of energy conserving speed determined by an algorithm vs the optimal algorithm.

	LO-criticality (Speedup Bound)	**HI-criticality (Speedup Bound)**	**LO-criticality (Approximation Ratio)**	**HI-criticality (Approximation Ratio)**
Optimal algorithm	P	1	P	1
An algorithm	sP	s	αP	1

Performance Evaluation

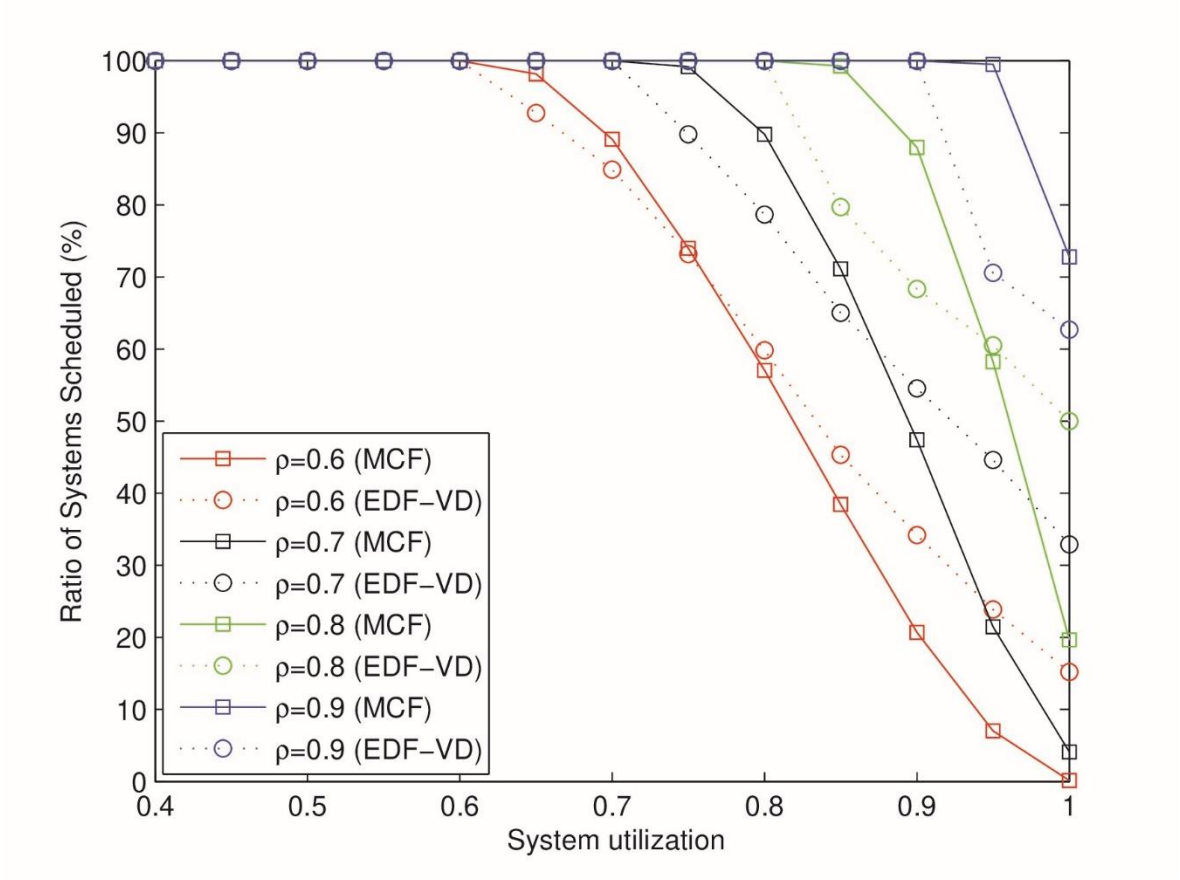
- **Speedup Bound, s.** How much faster processors are required for an algorithm to schedule the same taskset, scheduled by an optimal algorithm.
- **Approximation Ratio, α.** The ratio of energy conserving speed determined by an algorithm vs the optimal algorithm.

$$\textbf{Theorem 3.4.}\ (EDF-VD), s \leq (1/\min(U_L^L + U_H^H, U_L^L + \frac{(1-U_L^L)U_H^L}{(1-U_H^H-U_L^L)}))$$

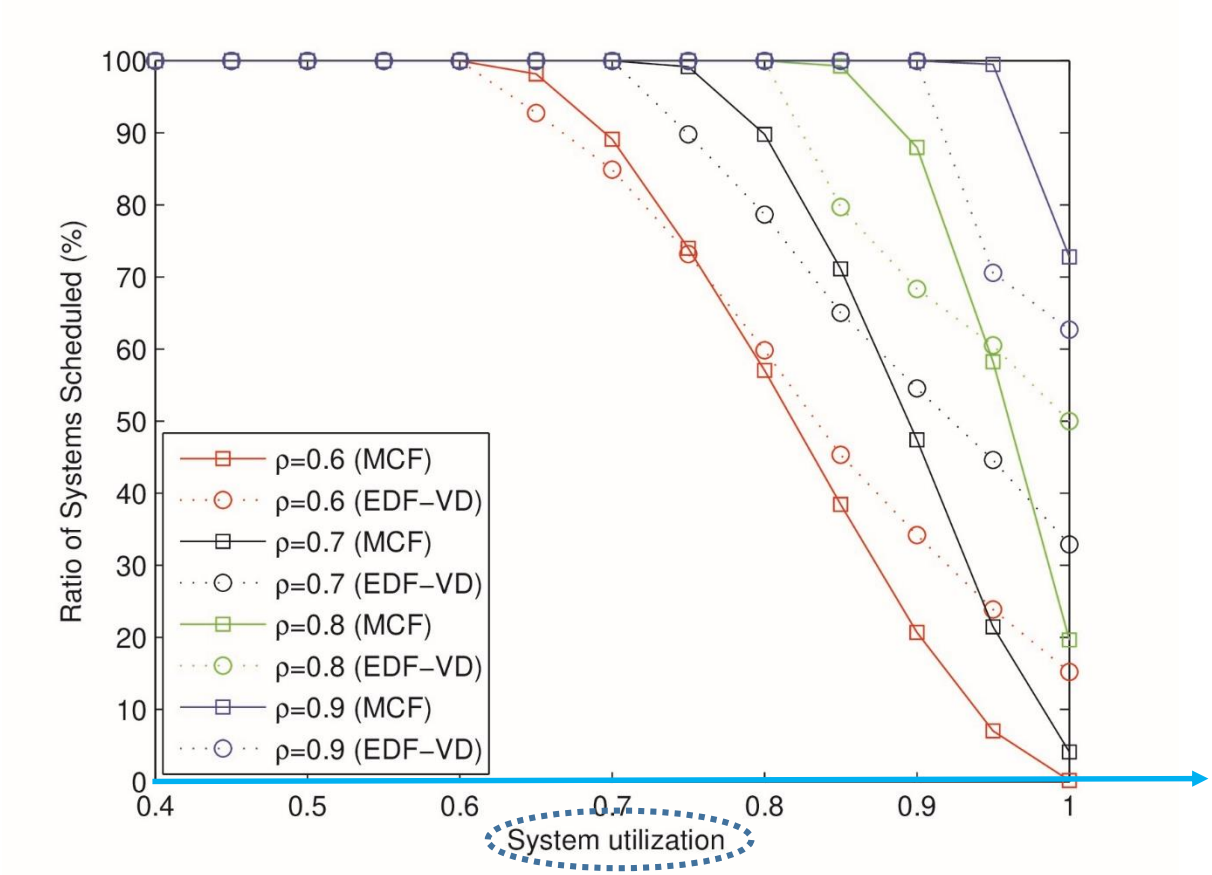
$$\textbf{Theorem 3.6.}\ (EDF-VD), \alpha \leq 1 + \frac{{U_H}^L(1-U_L^L)}{{U_L}^L(1-U_L^L-U_H^H)}$$

$$\textbf{Theorem 4.3.}\ (MCF), \alpha \leq \frac{1}{1+U^L-U^H}$$

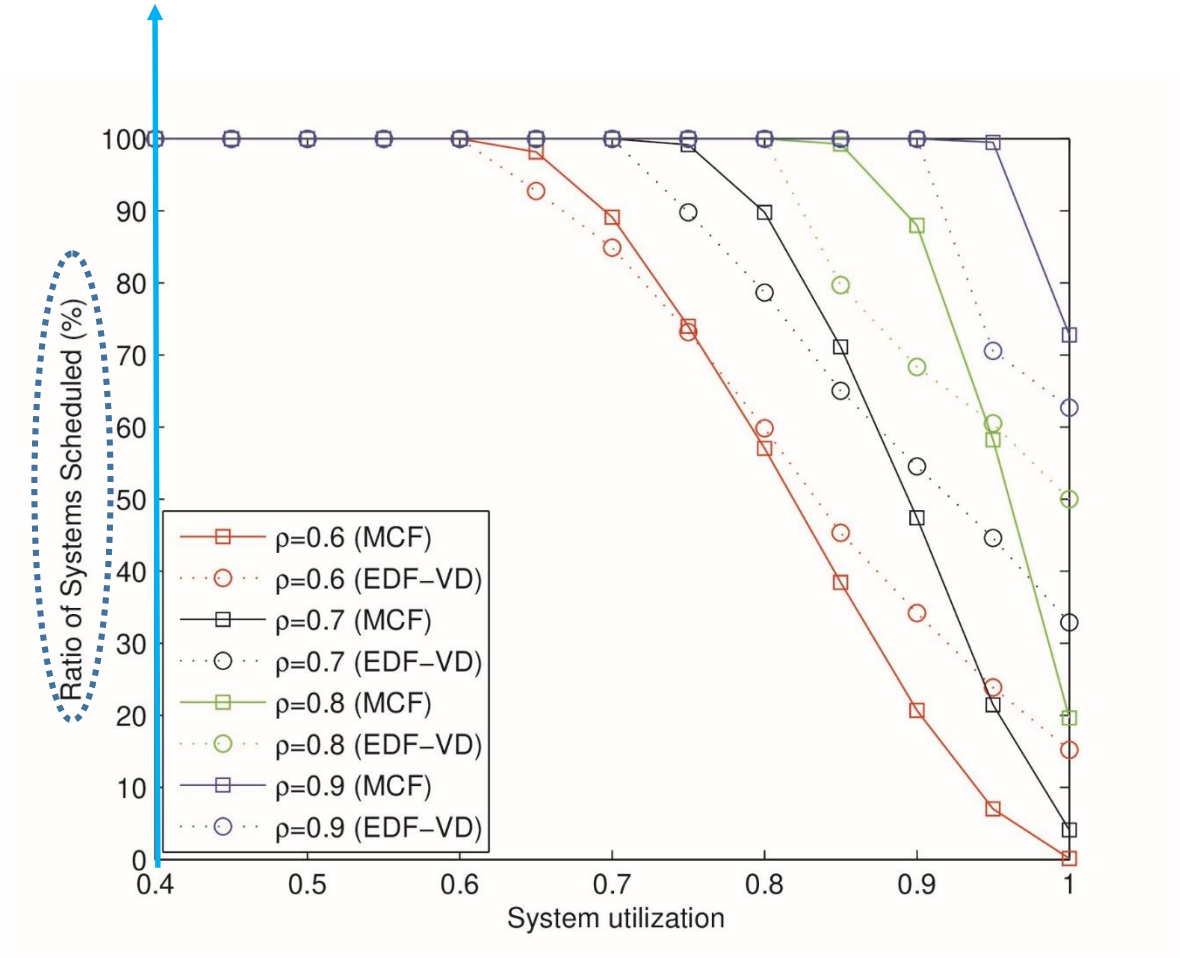
Simulation (EDF-VD vs MCF)



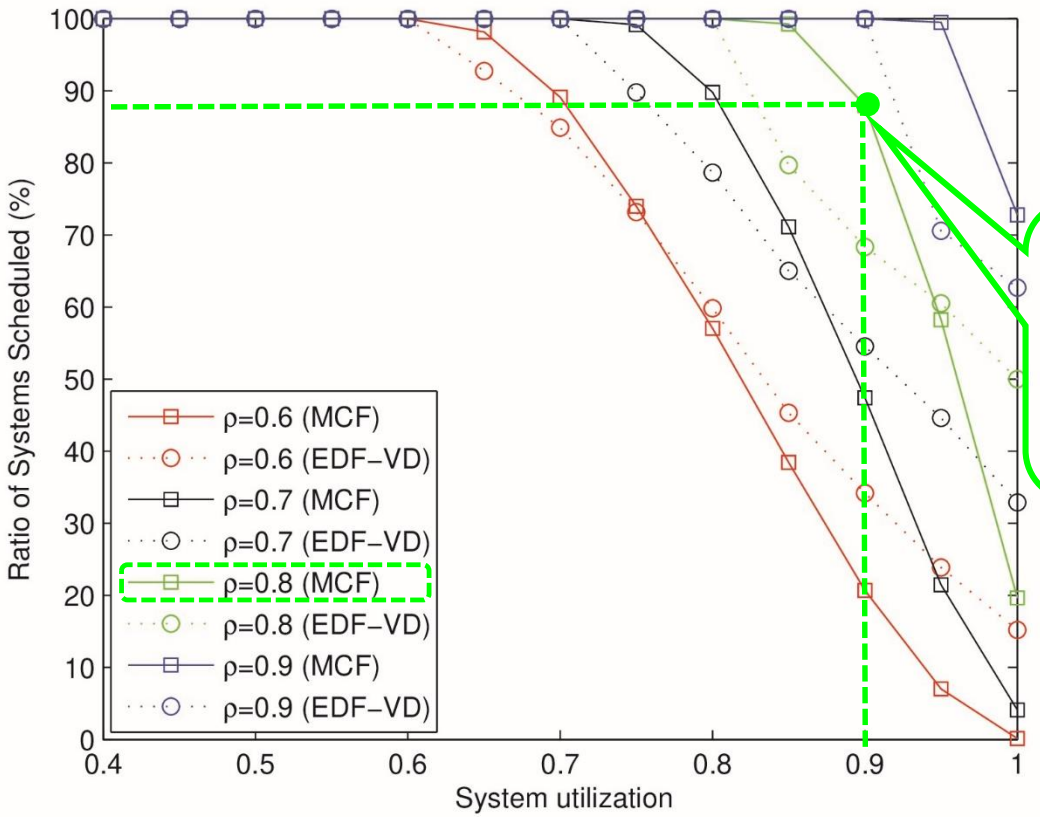
Simulation (EDF-VD vs MCF)



Simulation (EDF-VD vs MCF)

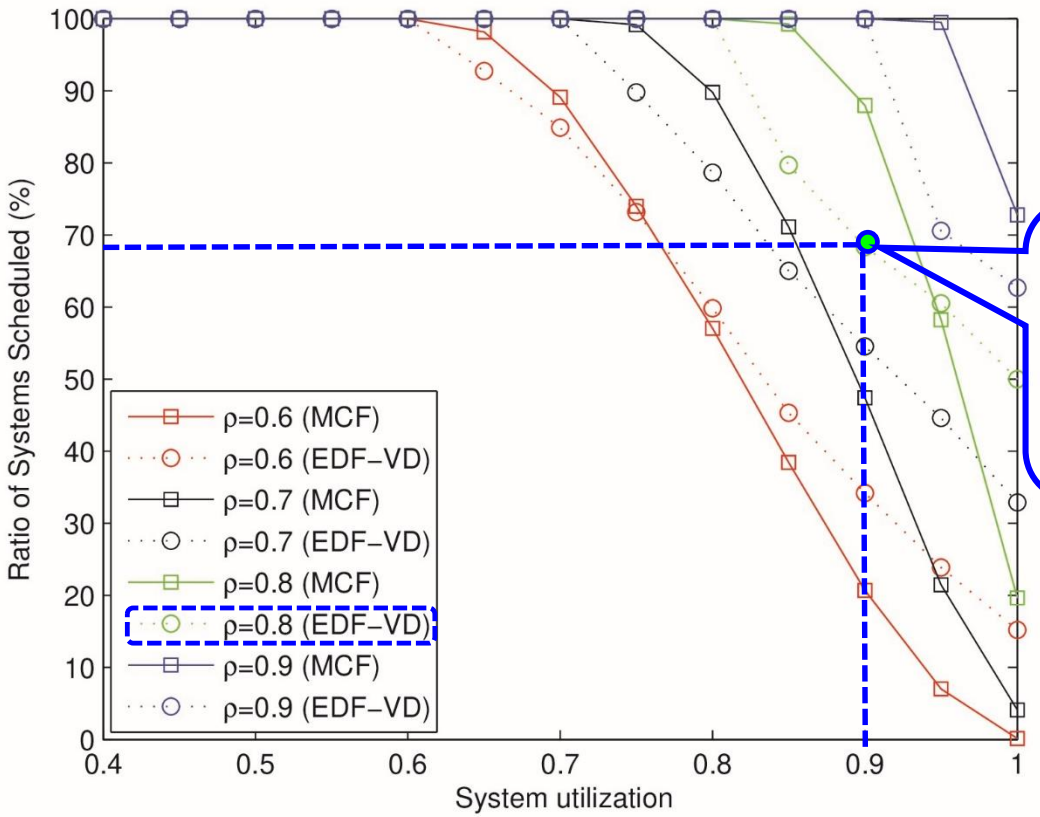


Simulation (EDF-VD vs MCF)



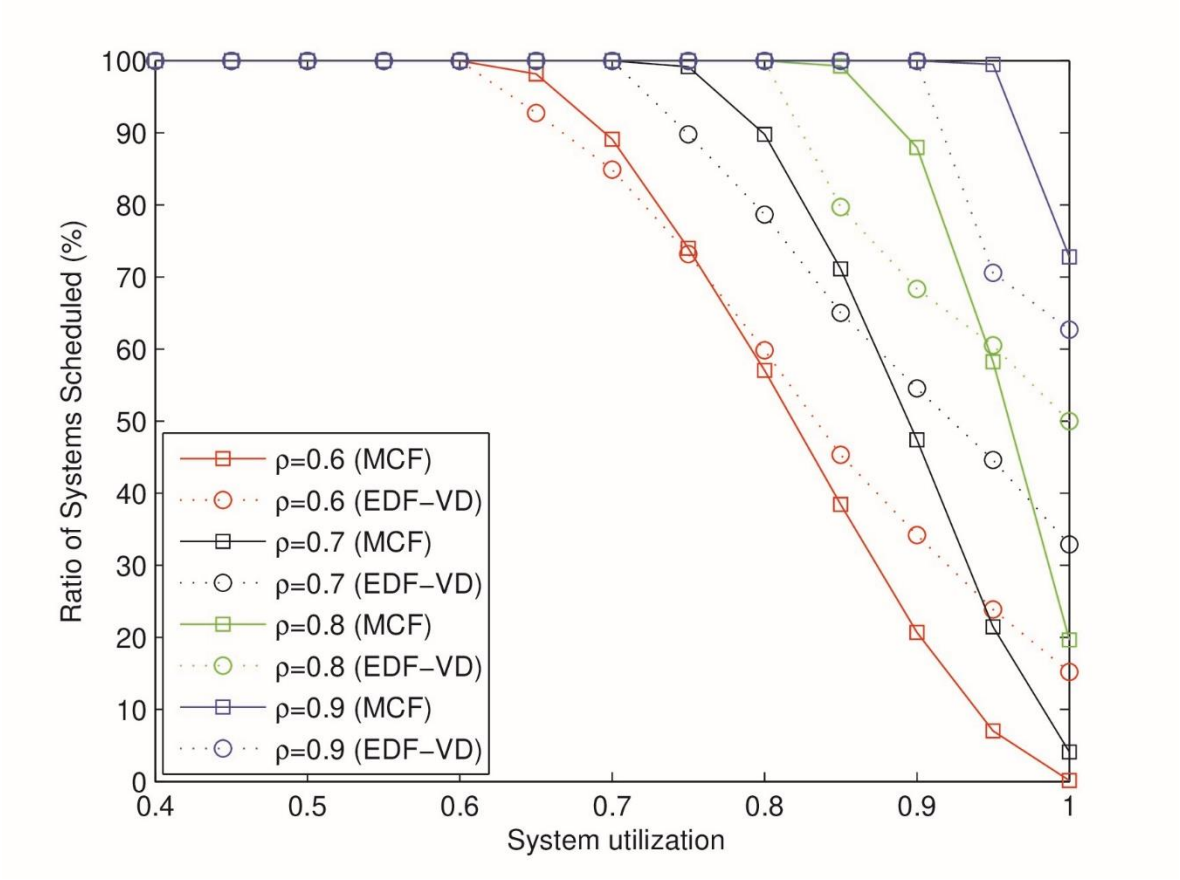
If $\rho = 0.8\ and$ system utilization is 0.9, MCF can schedule 88% of the randomly generated tasks.

Simulation (EDF-VD vs MCF)



If $\rho = 0.8\ and$ system utilization is 0.9, EDF-VD can schedule 68% of the randomly generated tasks.

Simulation (EDF-VD vs MCF)



Conclusion

- Energy-aware scheduling of MC tasks is challenging
- This work
  - Developed an integrated model combining precise scheduling of LO-criticality tasks on an energy-conserving platform.
  - Proposed schedulability tests under the EDF-VD and MCF scheduling framework.
  - Derived the speedup bound for EDF-VD and the approximation ratio for EDF-VD and MCF.

Conclusion

- Energy-aware scheduling of MC tasks is challenging
- This work
  - Developed an integrated model combining precise scheduling of LO-criticality tasks on an energy-conserving platform.
  - Proposed schedulability tests under the EDF-VD and MCF scheduling framework.
  - Derived the speedup bound for EDF-VD and the approximation ratio for EDF-VD and MCF.
- Future Goal
  - Considering multiprocessor platform and parallel task model.
  - Experimental evaluation on a real platform.

Thank You