A Switch-back Protocol for Task-level Criticality Mode on Mixed-Criticality Systems

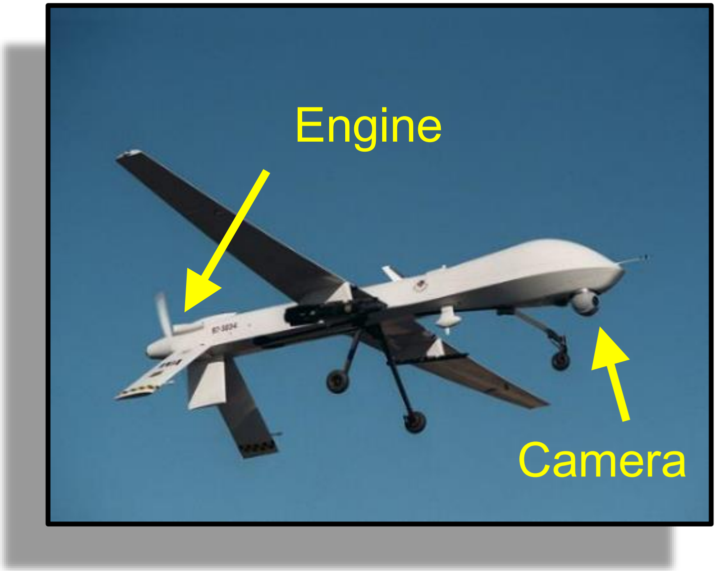
RTSOPS 2018, July, 3

Jaewoo Lee*, Hyungboo Baek^, Jinkyu Lee^

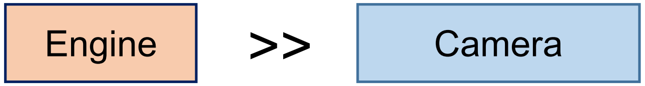
* Chung-Ang University, Korea

^ SungKyunKwan University (SKKU), Korea

Mixed-Criticality (MC) Systems

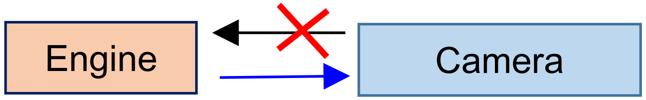


- MC systems: systems w/ functionalities of different criticalities
  - E.g., UAV



Engine >> Camera

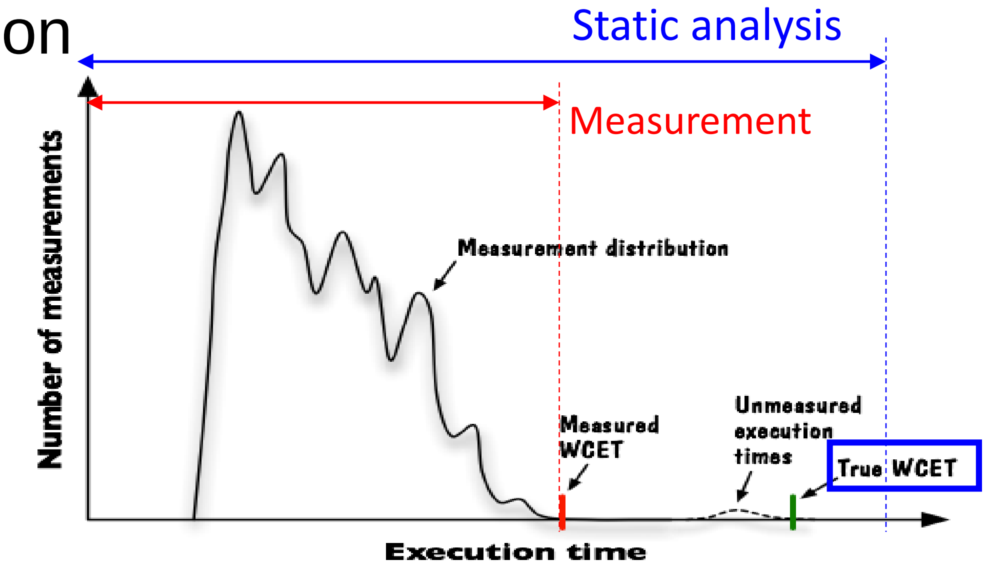
  - Practice: US FAA[1] adopted DO-178B
- The goal of MC systems
  - The correctness of HI-crit comp. is independent from LO-crit comp.



Engine Camera

MC Scheduling: Execution Times

- The Worst-Case Execution Time (WCET) of a task
  - Hard to find true WCET
    - Optimistic WCET
    - Pessimistic WCET



- MC task model [V07]
  - A task has multiple WCETs by different method to determine
  - To check a low-critical task (eg, camera), optimistic WCETs are used
  - To check a high-critical task (eg, engine), pessimistic WCETs are used

[V07] Vestal, **Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance**, RTSS, 2007

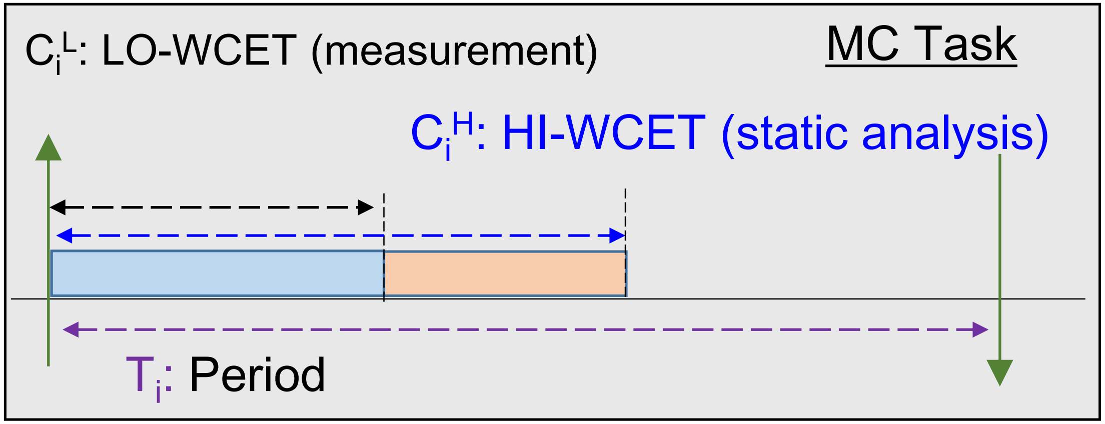
MC Scheduling: MC Task Model

- Dual criticality-levels
  - HI-criticality (safety-critical) and LO-criticality (normal)

- A MC task set: *n* MC tasks
  - MC task τ_i=(T_i, C_i^L, C_i^H, X_i

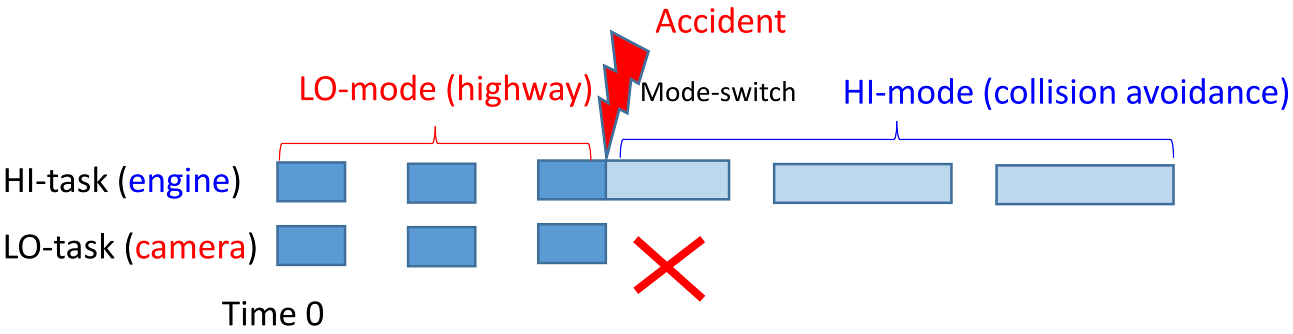
Task criticality (HI or LO)

HI-task (high-critical task, e.g., engine): X_i = HI
LO-task (low-critical task, e.g., camera): X_i = LO


C_i^L: LO-WCET (measurement)
MC Task
C_i^H: HI-WCET (static analysis)
T_i: Period

MC Scheduling: System Scenario

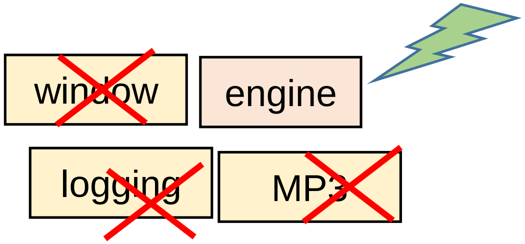
- System Mode: HI-mode (emergency) / LO-mode (normal)
- MC system is correct
  - LO-mode: all tasks with LO-WCETs are schedulable
  - HI-mode: only HI-tasks with HI-WCETs are schedulable
- MC system scenario (e.g., automobile)
  - Start in LO-mode
  - When exceeding LO-WCET (abnormal situation), *mode-switch* to HI-mode and drop all LO-tasks



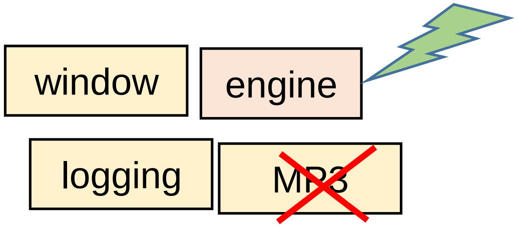
Recent MC scheduling

- Trends in MC scheduling
  - Earlier MC work **drops all low-criticality tasks (LO-tasks)** at mode switch
  - Recent MC scheduling work provides the degraded service for LO-task after mode switch
    - Degraded parameter (period, exec.) or selective task dropping
    - We consider to drop less jobs of LO-tasks

An example of automotive systems



Traditional MC scheduling

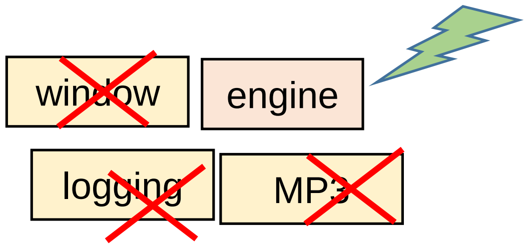


Recent MC scheduling

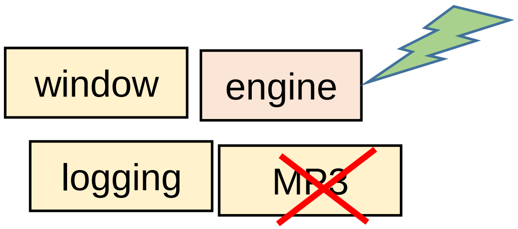
Recent MC scheduling

- Challenges
  - MC scheduling in frequent mode-switch situation?
    - Early work: HI-mode is a very rare event
    - What if mode switch is common event?
    - How to minimize the time length of HI-mode?
  - How to minimally drop jobs of LO-tasks?
    - Minimize the dropping of LO-tasks
    - Minimize the time length of HI-mode

An example of automotive systems



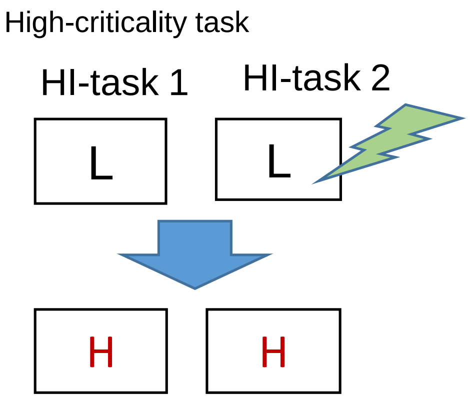
Traditional MC scheduling



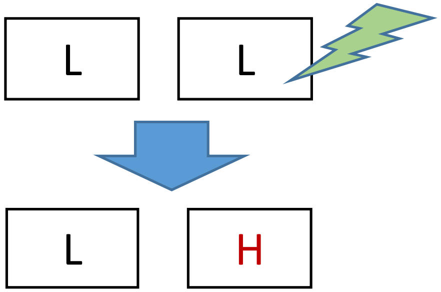
Recent MC scheduling

Criticality Mode

- System-level Criticality Mode
  - Assume all HI-crit. behaviors simultaneously
  - Drop all (or many) LO-tasks
  - Difficult to switch back



- Task-level Criticality Mode
  - Assume each HI-crit. behavior independently
  - Drop minimal LO-tasks
  - Easy to switch back



EDF-AD for task-level crit. mode

- **EDF-AD [L17]:** At mode switch, drop LO-tasks by an online test
- Algorithm description:
  - Schedule a HI-task with VD ($=xT_i$) in its LO-mode
    - VD →reserve room for additional exec. (HI-WCET – LO-WCET)
  - Drop LO-tasks selectively by an online test
- EDF-AD online schedulability test (only look current task state):

VD coefficient (0<x≤1)

$$\frac{U_{H1}^{L}}{x} + U_{L1}^{L} + U_{H2}^{H} + x \cdot U_{L2}^{L} \leq 1$$

τ_{H1}: LO-mode HI-tasks

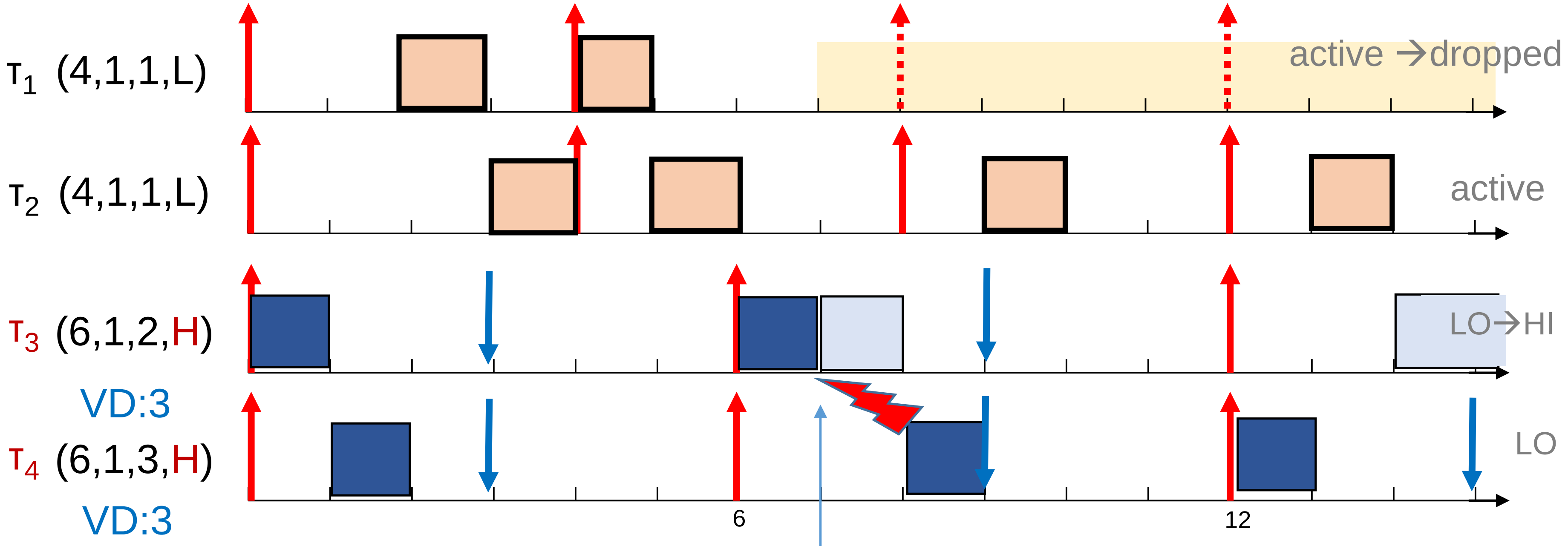
τ_{L1}: active LO-tasks

τ_{H2}: HI-mode HI-tasks

τ_{L2}: dropped LO-tasks

[L17] Lee et al., **MC-ADAPT: Adaptive Task Dropping in Mixed-Criticality Scheduling**, EMSOFT, 2017

Scheduling Example



x= 0.5

By online test, drop τ_1

$$U_{L1}^{L} + x \cdot U_{L2}^{L} + \frac{U_{H1}^{L}}{x} + U_{H2}^{H} \le 1$$

↑ job release

↓ job (virtual) deadline

Challenges for Switch-back Protocol

- How to switch back in task-level crit. mode?
  - Return to LO-mode when executing <= LO-WCET?
- How to resume LO-task activation?
  - At switch-back, we can restart the release of the dropped LO-tasks
  - Naïve resuming → deadline miss of HI-tasks
    frequent drop/resume
  - Need to resume LO-tasks based on some condition

Our Approach

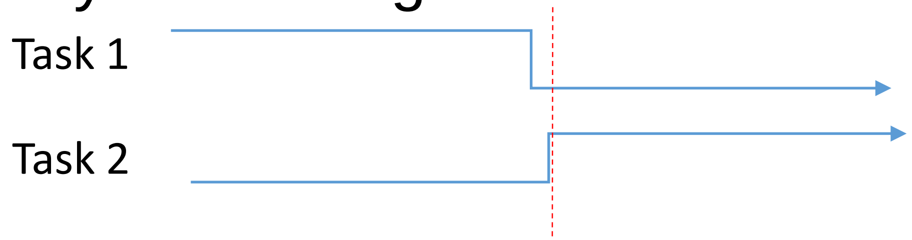
- Consider time-locality of over-executing HI-tasks
  - To avoid the fluctuation of criticality mode, set threshold # to switch mode
    - Ex) the threshold value = 3 → switch to LO-mode if the task executes less than LO-WCET 3 consecutive times.

- Resume Protocol
  - Resume the dropped LO-tasks based on the cond. (in progress)

$$\frac{U_{H1}^{L}}{x} + U_{H2}^{H} + U_{L1}^{L} + x \cdot U_{L2}^{L} \leq 1$$

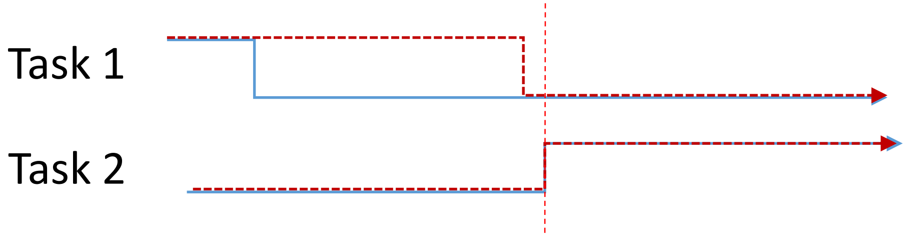
One problem for online test

- The adv. of online test in EDF-AD (no resume)
  - Only look at current status of tasks (no runtime history)
- Online test must change w/ resume
  - May resume right before mode switch



Task 1 is regarded as HI-mode? LO-mode?

- One approach: virtual mode
  - Virtually return after waiting VD



Conclusion

- Challenges for MC scheduling
  - MC scheduling for frequent mode switch?
  - How to minimally drop jobs of LO-tasks?
- Approaches: task-level criticality mode
  - At task-level mode-switch, EDF-AD drops minimal LO-tasks by online test
  - Develop switch back protocol for task-level crit. mode
- Problem
  - For a task, how to switch back from HI to LO?
  - At switch back situation, how to resume LO-tasks?
    - How to develop online test for the resuming?

Thank you

Questions & Comments?