A novel sufficient schedulability analysis for for floating defer preemption

Supervisor: Dr. Nguyen Thi Huyen Chau

Student : Vo Anh Hung

Outline

1. Overview
   1. Studied problem
   2. Background knowledge
2. Contributions
   1. The inexactitudes in [2].
   2. Corrected schedulability test.
   3. Novel sufficient schedulability test.
3. Conclusion and perspective

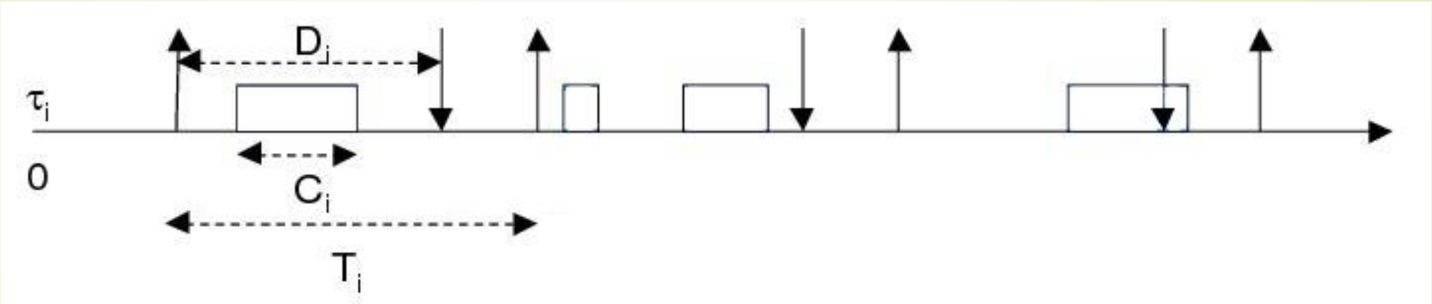
What is a real-time system?

- A computing system that processes information and produces output within precise time constraints.
- Quality of these systems depends on the validity of the output and the moment this result is produced.

➔ Importance of the schedulability tests.

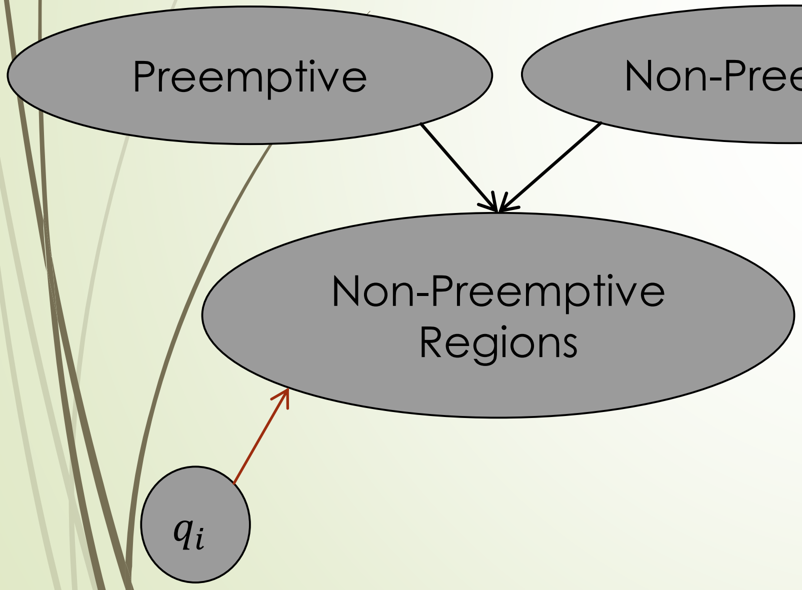
Basic notions

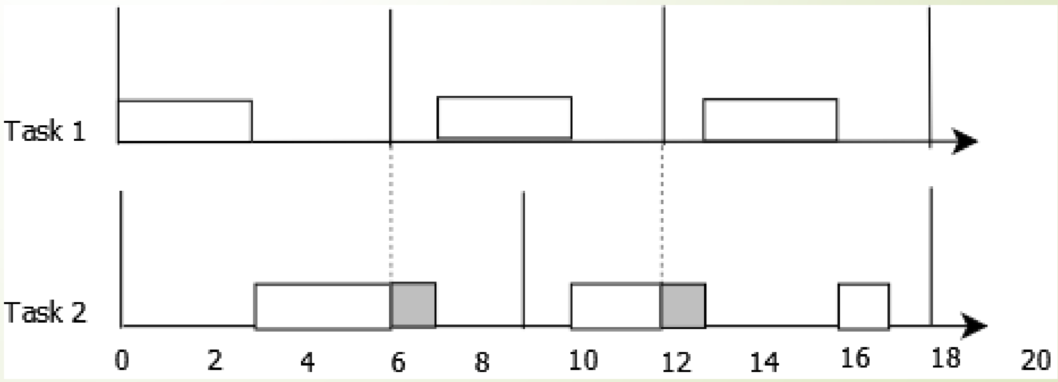
- n: number of tasks.
- τ_i: The i^{th} task, each task can perform infinite times (job $\tau_{i,k}$).
- Each task τ_i consists of three basic parameters:
  - C_i: the worst-case execution time
  - T_i: period
  - D_i: relative deadline
- Constrained deadline: The deadline of any task smaller than the period.
- Arbitrary deadline: The deadline of any task may be greater than the period.



Scheduling policies

- Fixed priority scheduling: among ready tasks, CPU will be assigned to the highest priority one.





Principle of schedulability analysis

- Schedulability verification: only sufficient or exact tests.
- Principle: Always test the system in the worst-case scenario.
  - If passes the test, the system is schedulable.
  - Otherwise, the system is unschedulable.
- Critical instant: The system phase that produces the longest task response time.

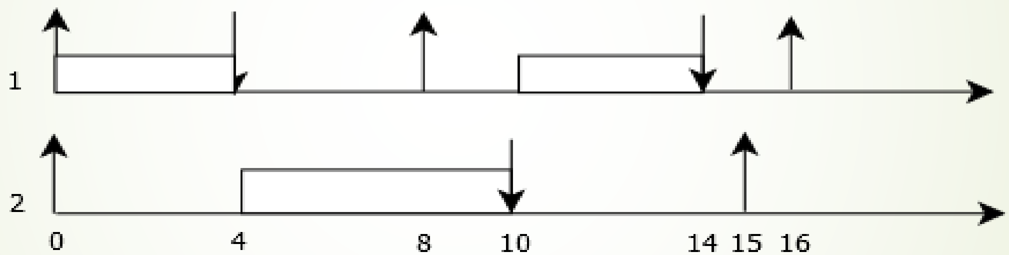
➔ Critical instant is an important factor to verify the schedulability in case that the system phase in unknown.

Critical instant in [2] - revisited

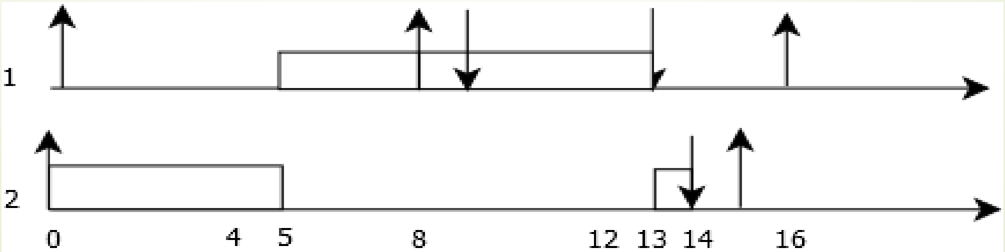
- The critical instant for P, NP (1):
  - Simultaneously released with all of its higher priority tasks.
  - Experiences its largest blocking time.
- [2] has claimed that (1) also defines the critical instants for NPR tasks.
- The thesis has proved that this statement is not correct by a counter-example.

Critical instant in [2]– counter-example

Task	C	D	T	q
1	4		8	0
2	6		15	5



When $\phi_1 = \phi_2, R_2 = 10$



When $\phi_1 - \phi_2 \downarrow 0, R_2 = 14$

Schedulability test in [2] - revisited

- [2] has claimed that:

  A task set τ with floating non-preemptive regions is schedulable with a fixed priority algorithm <u>if and only if</u> $\forall \tau_i \in \tau, \exists t \in TS(\tau_i)$ such that:

  $$W_i(t) + B_i \leq t$$

- The thesis has proved this to be incorrect by a counter-example.
- The corrected test:

  A task set τ with floating non-preemptive regions is schedulable with a fixed priority algorithm <u>if</u> $\forall \tau_i \in \tau, \exists t \in TS(\tau_i)$ such that:

  $$W_i(t) + B_i \leq t$$

A novel sufficient schedulability test for NPR with arbitrary deadlines

- Extend the corrected test for arbitrary deadlines:

Theorem: *A task set T with non-preemptive regions and aribitrary deadlines is schedulable if:*

$$\forall t_i \in T, \forall k \in N: 0 < k \leq l_i, \exists t \in S_{i,k}: \\ W_{i,k}(t) + B_i \leq t$$

Where:

$$\begin{cases} S_{i,k} = \{aT_j | j < i, \frac{(k-1)T_i}{T_j} < a \leq \frac{(k-1)T_i + D_i}{T_j}\} \\ W_{i,k}(t) = kC_i + \sum_{j<i} RFB_j(t) \end{cases}$$

Conclusion and perspective

- Conclusion:
  - Present some inexactitudes in [2].
  - Correct the schedulability test in [2].
  - Propose a novel sufficient schedulability test for a more general context.
- Perspective:
  - Will refine all the other results in [2].
  - Will characterize the critical instant to propose a necessary and sufficient condition for verifying the system schedulability in NPR.

References

1. *[1] R. Bril, J. Lukkien, and W. Verhaegh. Worst-case response time analysis of realtime tasks under fixed-priority scheduling with deferred preemption. Real-Time Systems, 42(1-3):63–119, 2009.*
2. *[2] G. Yao, G. Buttazzo, and M. Bertogna. Bounding the maximum length of nonpreemptive regions under fixed priority scheduling. In Proceeding of the 16th IEEE international conference on embedded and Real-Time Computing Systems and Applications(RTCSA 2009), pages 351–360, China, 2009.*
3. *[3] G. C. Buttazzo. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. Springer, 2006.*

Thank you for your attending