43rd IEEE Real-Time Systems Symposium

9th Workshop on Mixed Criticality Systems

Mixed-Criticality Scheduling for Parallel Real-Time Tasks with Resource Reclamation

Qingqiang He[1], Nan Guan[2], Xu Jiang[3]

1 The Hong Kong Polytechnic University, China
2 City University of Hong Kong, China
3 Northeastern University, China

Summary

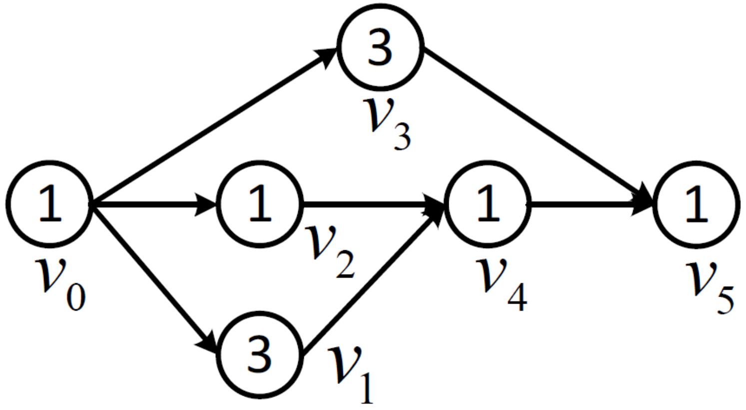
- federated scheduling of parallel real-time tasks
  - each task is scheduled independently on a set of dedicated cores
  - so we only need to consider the scheduling of one task on multi-cores

- target of our approach
  - guarantee the deadline for hard real-time tasks
  - reclaim computing resources for soft real-time tasks

- the proposed approach
  - online monitor the execution of hard real-time tasks
  - dynamically adjust the allocated number of cores for hard real-time tasks

The Parallel Real-Time Task

- for the scheduling
  - volume *vol(G)* : the total workload in this task
  - length *len(G)* : the workload in the longest path
  - these parameters can be measured without knowing the structure of the DAG
- for the analysis
  - the DAG task model

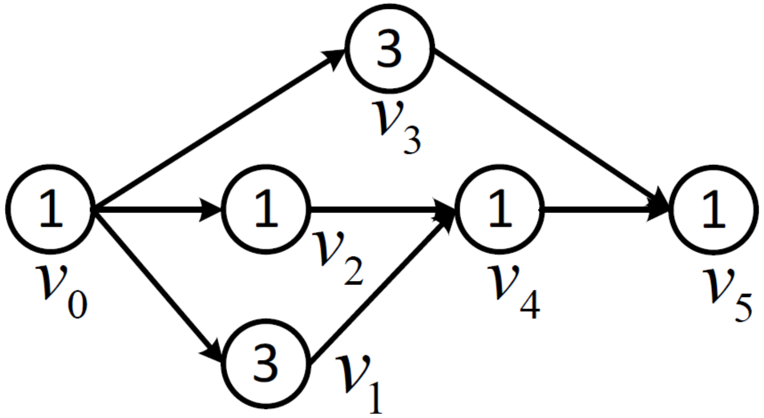


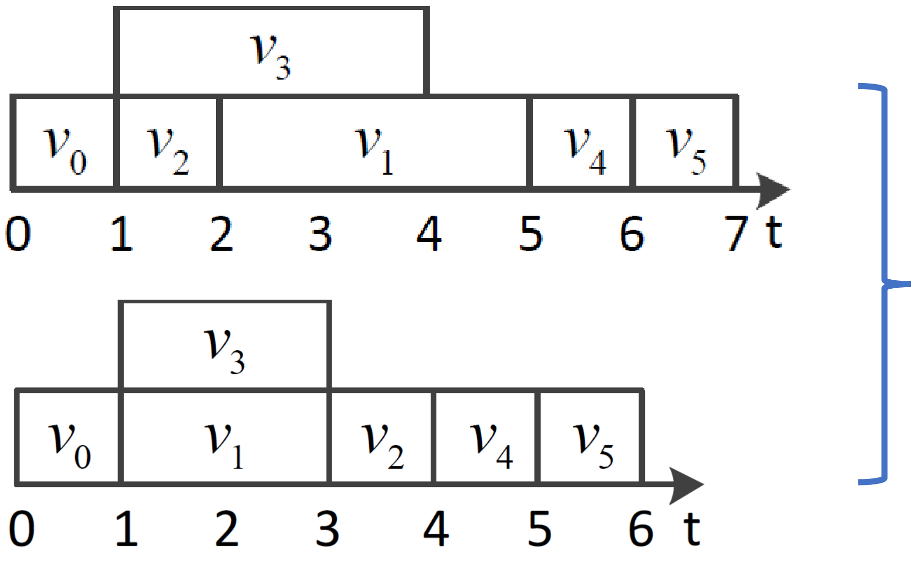
The longest path is $\lambda = (v_0, v_1, v_4, v_5)$

$len(G) = 6 \qquad vol(G) = 10.$

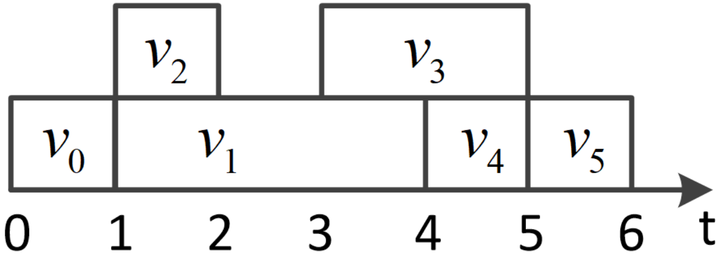
The Scheduling

- work-conserving
- identical multi-core platform





work-conserving



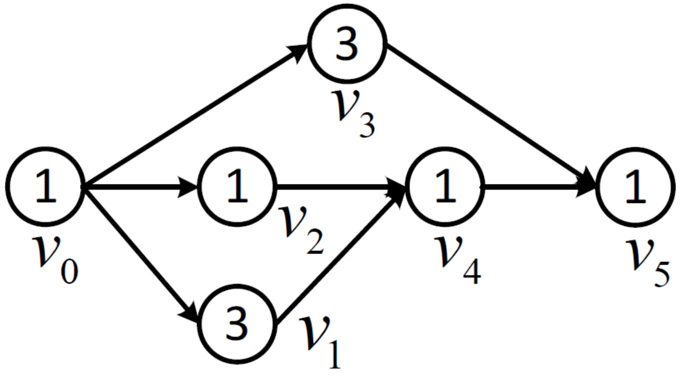
not work-conserving

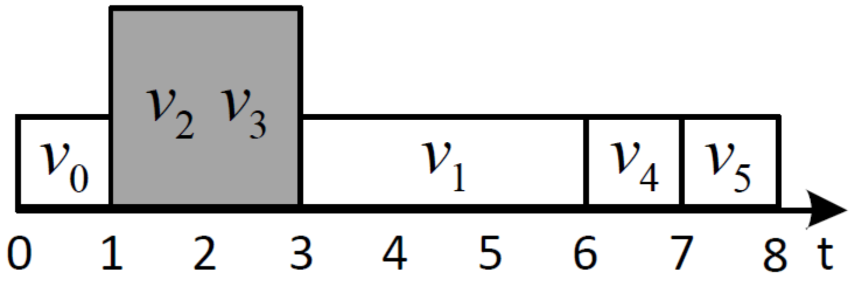
Motivation

- Graham’ bound in federated scheduling

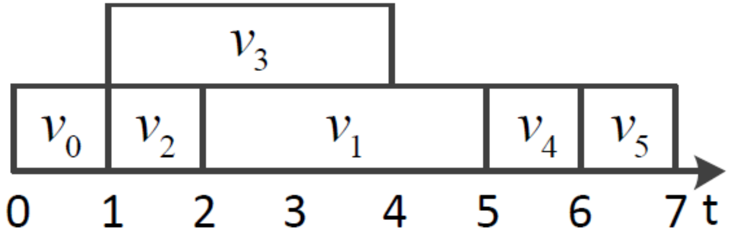
$$R \leq len(G) + \frac{vol(G) - len(G)}{m} \qquad m = \left\lceil \frac{vol(G) - len(G)}{D - len(G)} \right\rceil$$

- type 1 : analysis pessimism
  - assuming that vertices not in the longest path do not execute in parallel with the execution of the longest path.





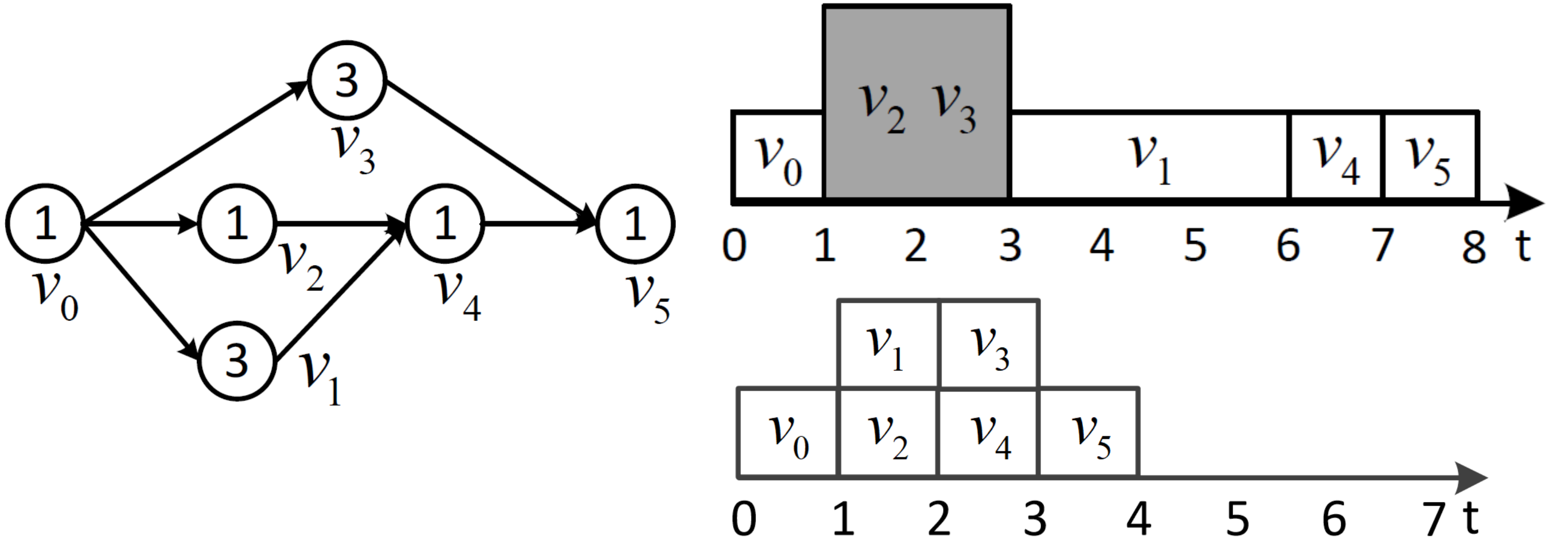
analysis assumption



a possible schedule

Motivation

- type 2 : execution pessimism
  - volume and length are based on the worst case execution time (WCET).
  - the actual execution time of vertices can be far less than the WCET



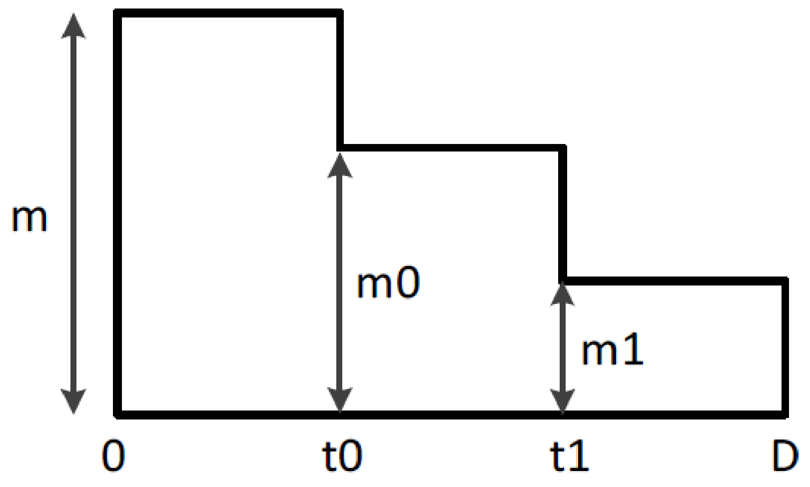
analysis assumption

an actual schedule

- due to these two types of pessimism, the federated scheduling can waste a large amount of computing resources.

The Proposed Approach

- online monitor the execution of hard real-time tasks
  - the volume of the executed workload $w(t_i)$
  - the length of time intervals during which at least one core is idle $l(t_i)$
- adjust the allocated number of cores for hard real-time tasks



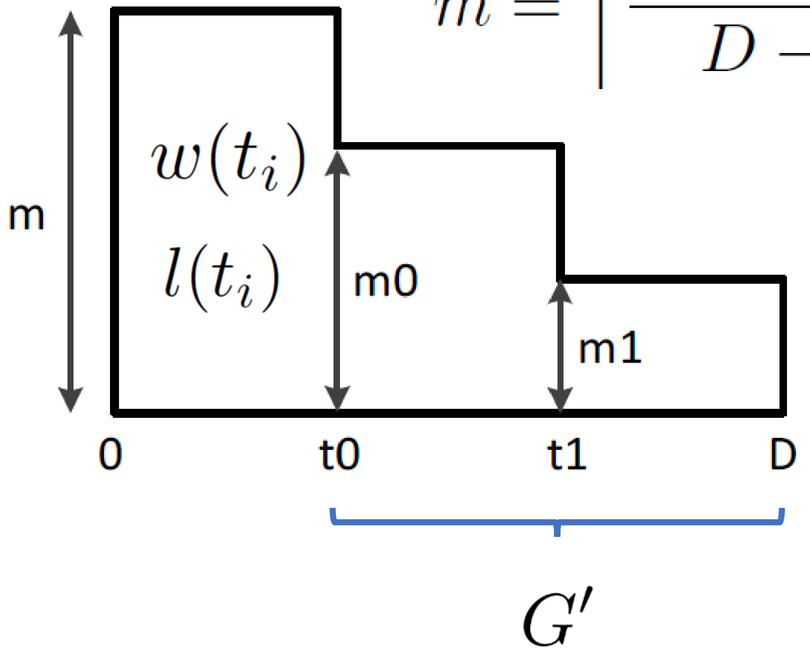
Definition 1 (Allocation Vector)**.** *For a parallel real-time task* (G, D, T), *the allocation vector* Φ *is a set of time points* $\{t_0, \cdots, t_k\}$ $(k \geq 0)$ *satisfying all of the following conditions.*

1) $\forall i \in [0, k]$, $0 \leq t_i < D$.
2) $\forall i, j \in [0, k]$ *and* $i < j$, $t_i < t_j$.

The Proposed Approach

- online monitor the execution of hard real-time tasks
  - the volume of the workload executed $w(t_i)$
  - the length of time intervals during which at least one core is idle $l(t_i)$
- adjust the allocated number of cores for hard real-time tasks

$$m = \left\lceil \frac{vol(G) - len(G)}{D - len(G)} \right\rceil$$



$$vol(G') = vol(G) - w(t_i)$$

$$len(G') = len(G) - l(t_i)$$

$$D' = D - t_i$$

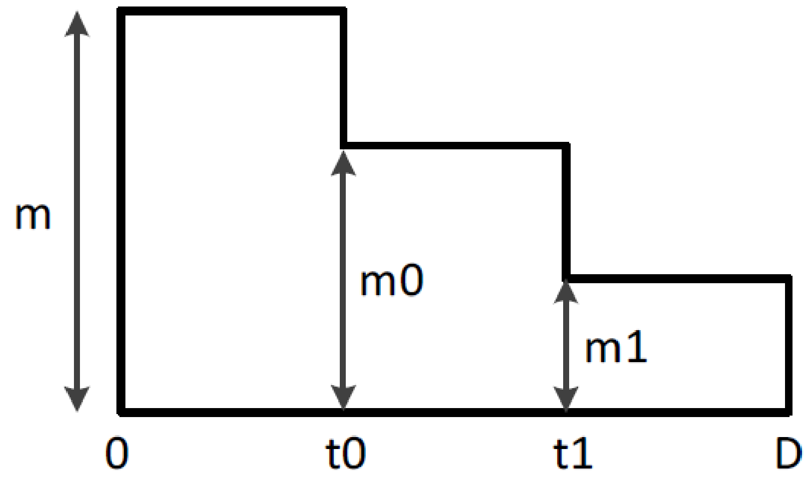
$$m_i = \left\lceil \frac{vol(G') - len(G')}{D' - len(G')} \right\rceil$$

The Proposed Approach

- the critical path
  - the length of time intervals during which at least one core is idle $l(t_i)$
  - the length of the remaining graph is bounded by

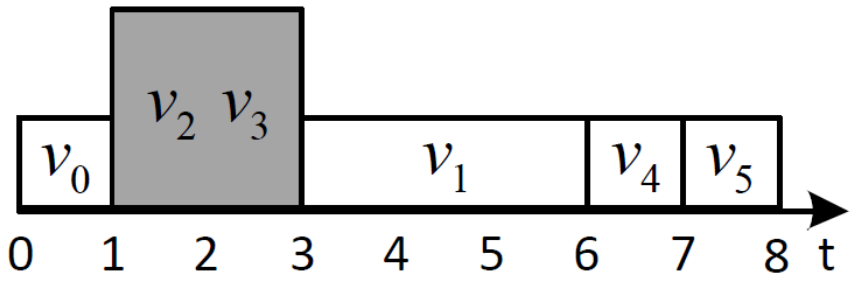
$$len(G') = len(G) - l(t_i)$$

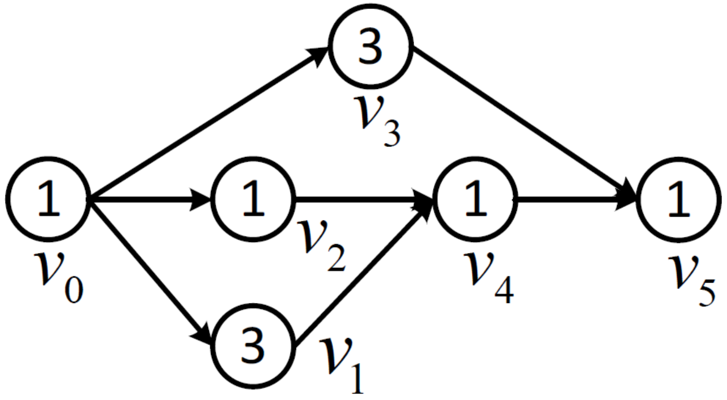
- our approach dominates the original federated scheduling
  - the adjusted number of cores cannot increase (Corollary 1)

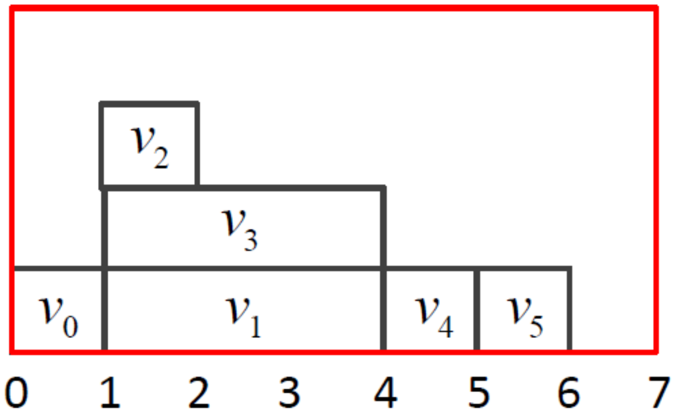


The Proposed Approach

- design principle for soft real-time tasks
  - how to determine the allocation vector
- monitor the execution satisfying both of the following conditions
  - at least one core is idle
  - more than one core are busy



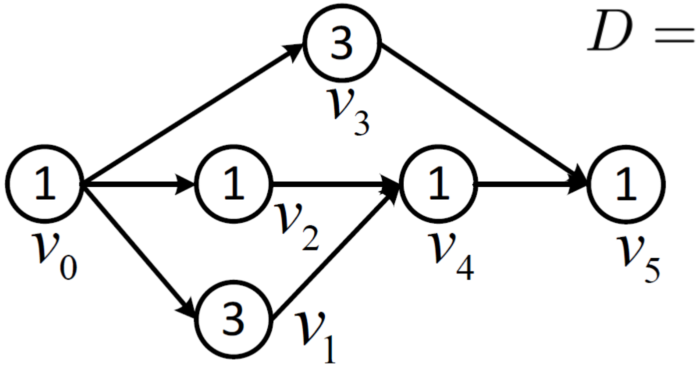




An Example

$len(G) = 6 \quad vol(G) = 10$

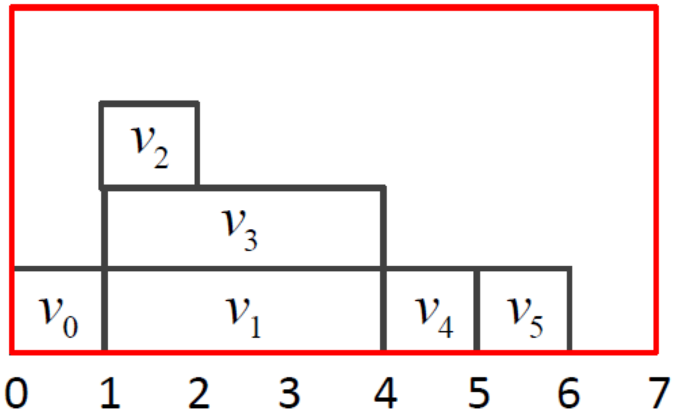
$D = 7$



$$m = \left\lceil \frac{vol(G) - len(G)}{D - len(G)} \right\rceil = 4$$

$$m \times D = 4 \times 7 = 28$$

federated scheduling



allocation vector

$$\Phi = \{t_0 = 2, t_1 = 3\}$$

$t_0 = 2 \quad w(t_0) = 4 \quad l(t_0) = 2$

$$vol(G') = vol(G) - w(t_i) = 6$$

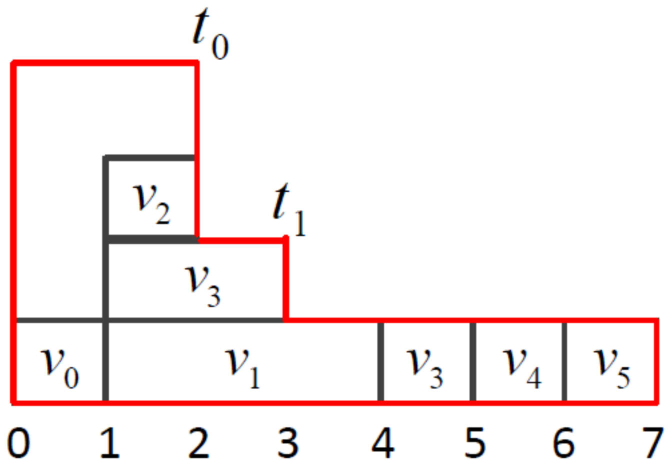
$$len(G') = len(G) - l(t_i) = 4$$

$$D' = D - t_i = 5$$

$$m_0 = \left\lceil \frac{vol(G') - len(G')}{D' - len(G')} \right\rceil = 2$$

$$4 \times 2 + 2 \times 1 + 1 \times 4 = 14$$

our approach



Conclusion

- the proposed approach
  - online monitor the execution of hard real-time tasks
  - dynamically adjust the allocated number of cores for hard real-time tasks
- allocation vector as interface
  - for hard real-time tasks, a schedulability test under the interface
  - for soft real-time tasks, the design principle of how to determine the interface to reclaim computing resources
- using an example to illustrate the effectiveness of the proposed approach.

Thank you and Questions?