Suspension-Based Locking Protocols for Parallel Real-Time Tasks

Xu Jiang, Nan Guan, Yue Tang, Weichen Liu, Hancong Duan

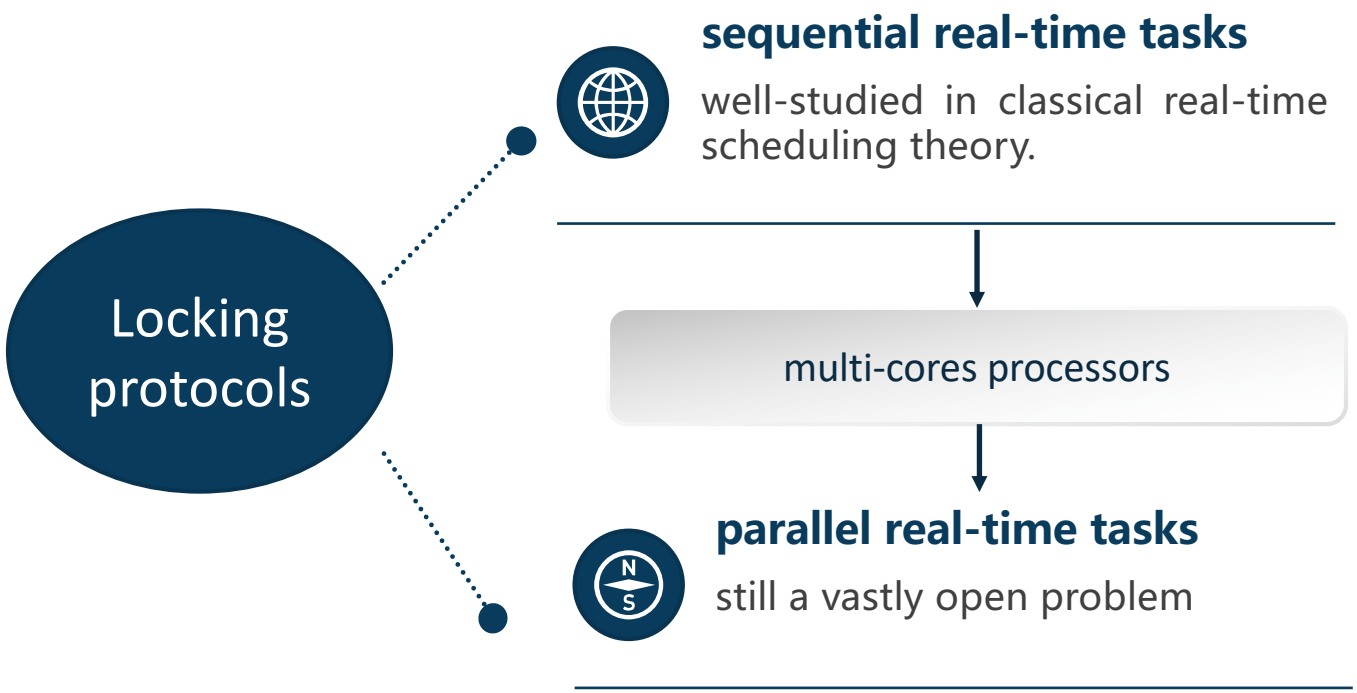
CONTENTS

1

PART ONE

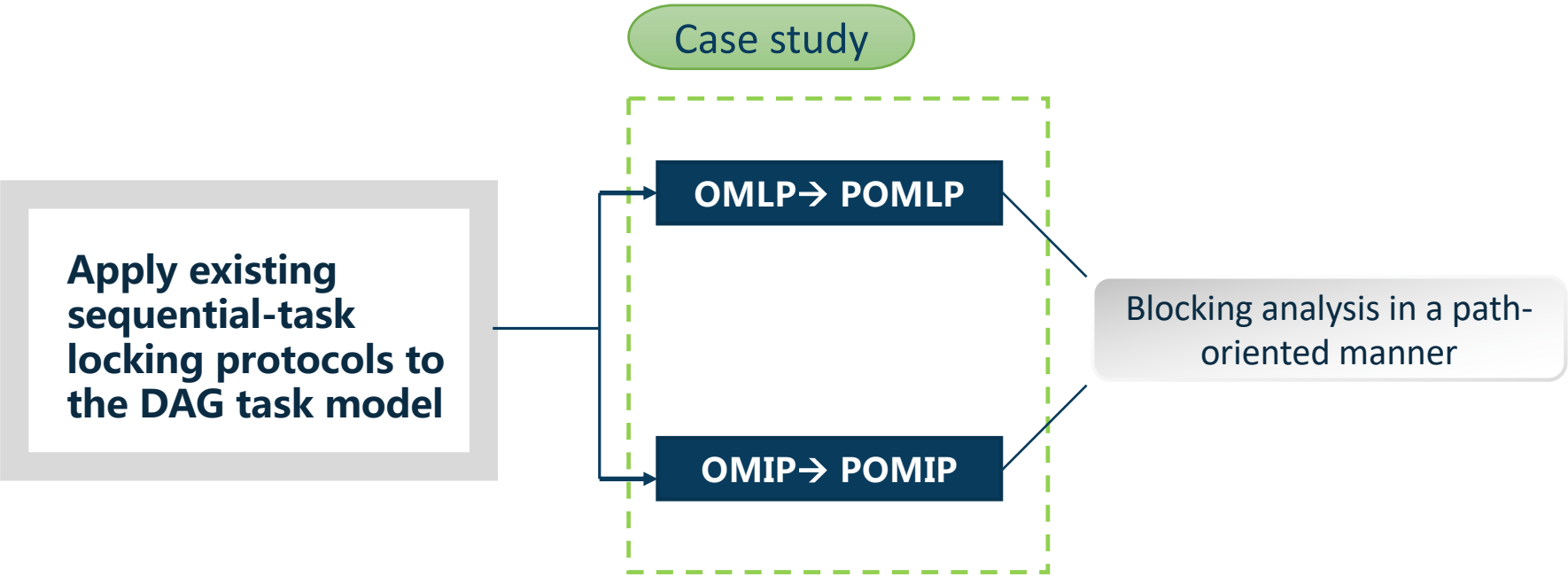
INTRODUCTION

Background


Locking protocols
sequential real-time tasks
well-studied in classical real-time scheduling theory.
multi-cores processors
parallel real-time tasks
still a vastly open problem

Main Work

In this paper, we study the design and analysis of suspension-based locking protocols (or called semaphores) for Directed Acyclic Graph (DAG) parallel real-time task models.

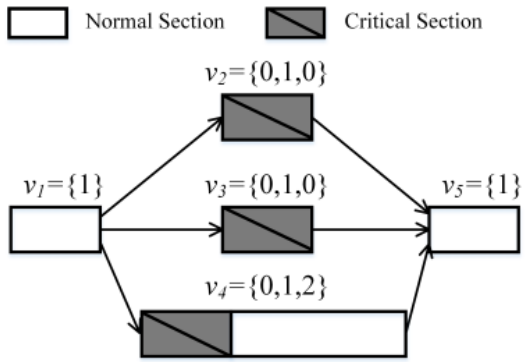


2

PART TWO

PRELIMINARY

SYSTEM MODEL



Task Model

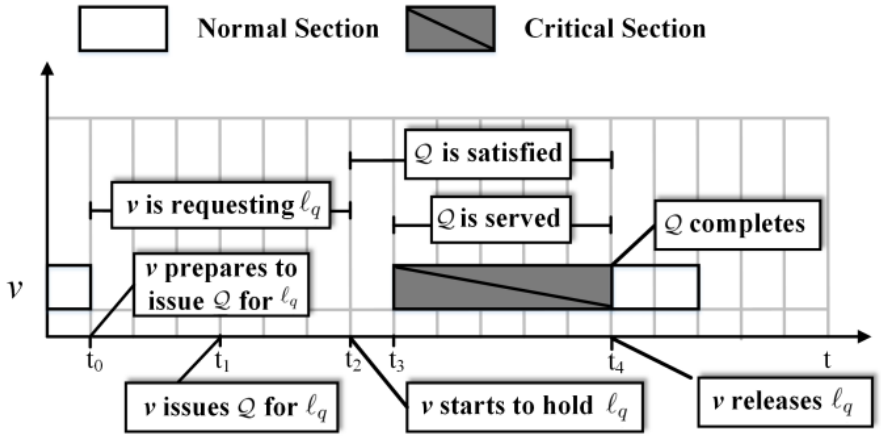
task set : $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_n\}$ platform : m processors

each task τ_i: $G_i = \langle V_i, E_i \rangle$ constrained deadline : $D_i \leq T_i$

each vertex : $v = \{c_{i,1}, c'_{i,1}, \cdots, c'_{i,s-1}, c_{i,s}\}$

Resource Model

$\Theta = \{\ell_1, ..., \ell_{n_r}\}$



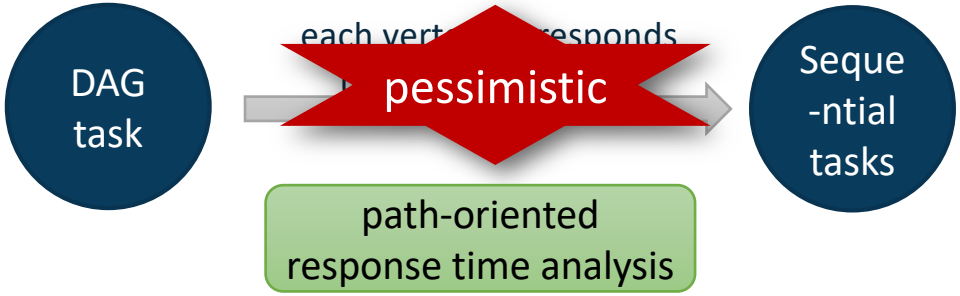
Scheduling

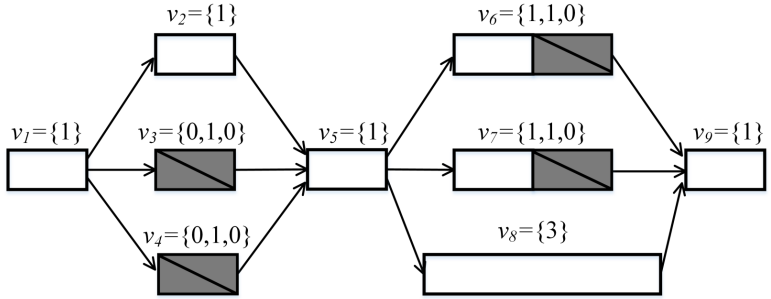
federated scheduling

work-conserving scheduling

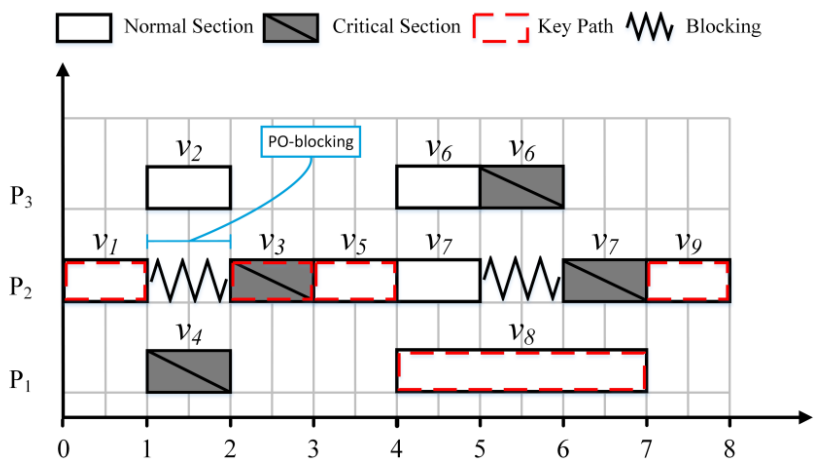
Main Challenge

The main challenge of applying existing locking protocols to DAG tasks is how to count the blocking time into the schedulability test for DAG tasks.





- Not all blocking time caused by resource contentions may interfere the schedulability of a DAG task.
- The schedulability of a DAG job is decided by the finish time of its latest finished vertex rather than the finish time of each individual vertex.
- Due to the precedence constraints, a lot of resource access contentions are impossible.



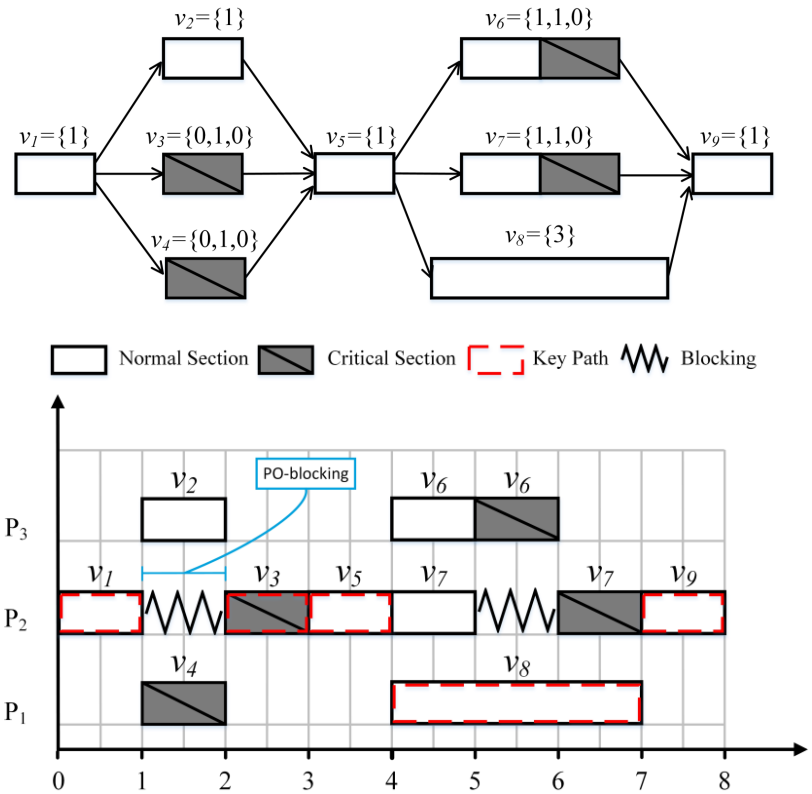
A Path-Oriented Response Time Analysis Framework

Key Path: A key path of job J_i is a complete path $\lambda = \{v_1, \cdots, v_p\}$ in G_i, s.t., $\forall j$: $1 < j \leq p$, v_{j-1} is the predecessor of v_j with the latest finish time among all predecessors of v_j.

$$m_i \times (f(J_i) - r(J_i)) = W_i(\lambda) + \mathcal{S}_i(\lambda) + \mathcal{I}_i(\lambda)$$

- **Working Time** $W_i(\lambda)$: the cumulative length of time on m_i processors spent on executing workload of J_i (either normal sections or critical sections).
- **PO-Blocking Time** $\mathcal{S}_i(\lambda)$: the cumulative length of time on m_i processors being idle when λ is suspended.
- **PO-Idle Time** $\mathcal{I}_i(\lambda)$: the cumulative length of time on m_i processors being idle when λ is not suspended

$$\mathcal{R}_i \leq \max_{\lambda \in \tau_i} \left\{ \frac{\mathcal{C}_i + (m_i - 1) len(\lambda) + \mathcal{S}_i(\lambda)}{m_i} \right\}$$



3

PART THREE

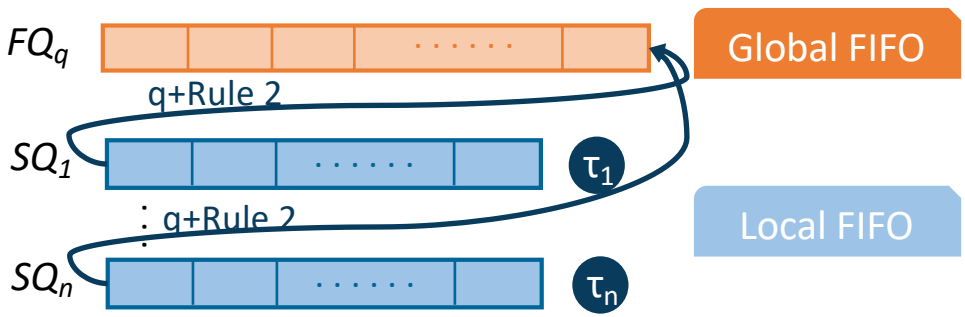
BLOCKING ANALYSIS

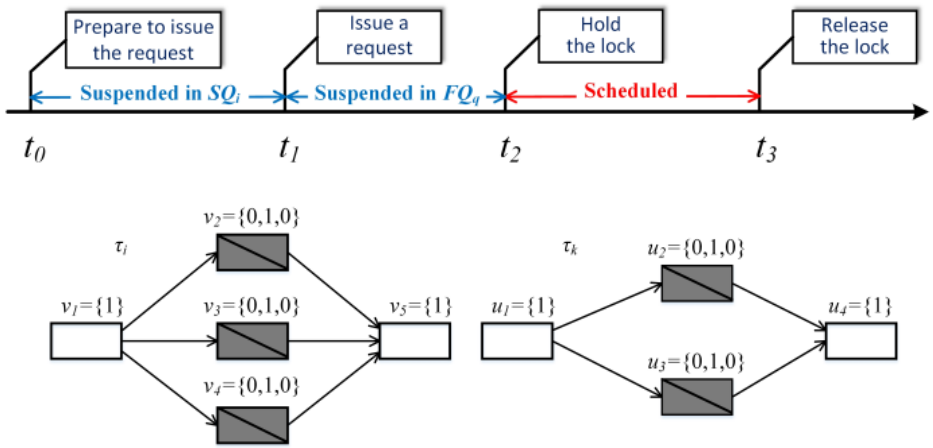
POMLP

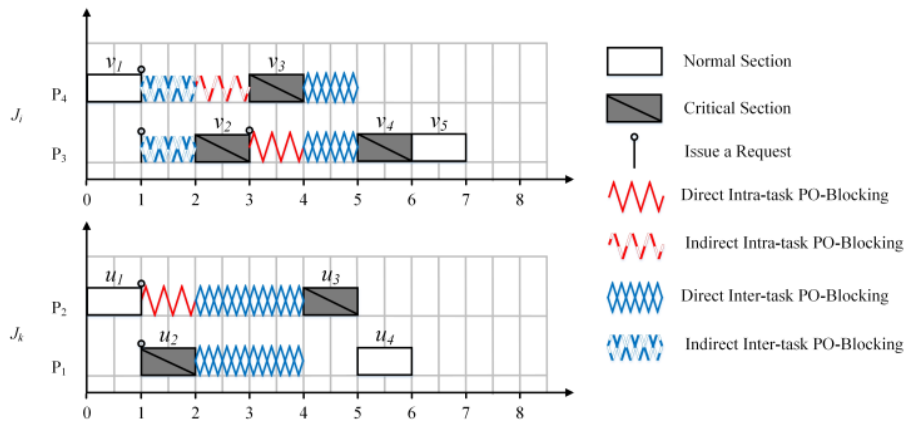
Rules

- **Rule 1:** A vertex holding a lock can preempt a scheduled vertex not holding a lock whereas a scheduled vertex holding a lock can not be preempted.
- **Rule 2:** At any time t, a vertex of J_i is allowed to issue a request iff the number of issued requests that do not complete (for all kinds of resources) of J_i is less than m_i.

Queue structure





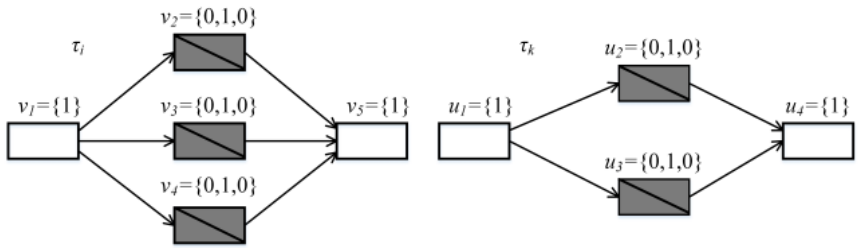


POMLP

PO-Blocking Time

$$\mathcal{S}_i(\lambda) = \sum_{\ell_q \in \Theta_i} \left(\mathcal{S}^I_{i,q}(\lambda) + \mathcal{S}^O_{i,q}(\lambda)\right)$$

Intra-task Inter-task



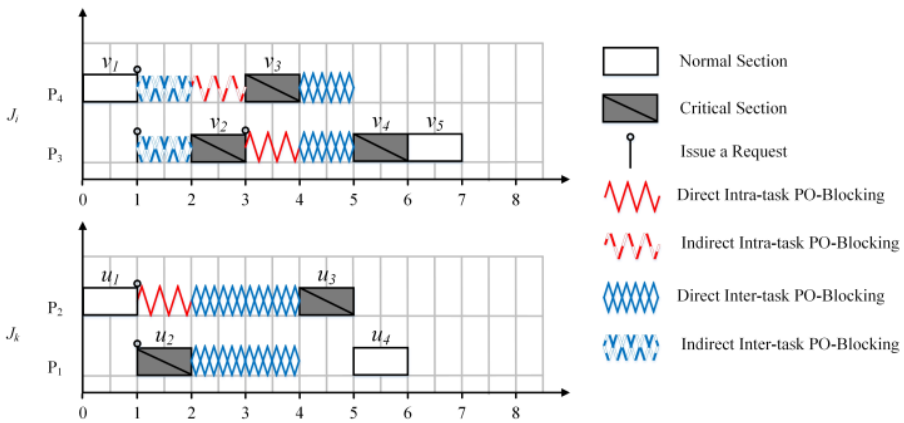
Lemma 5. *Under POMLP* $\mathcal{S}^I_{i,q}(\lambda) \le \mathcal{F}^I(N^\lambda_{i,q})$ *where*

$$\mathcal{F}^I(N^\lambda_{i,q}) = \begin{cases} (N_{i,q} - N^\lambda_{i,q})(m_i - 1)L_{i,q}, & N^\lambda_{i,q} > 0 \\ (\dfrac{\alpha(\alpha-1)}{2} + (m_i - 1)(N_{i,q} - \alpha))L_{i,q}, & N^\lambda_{i,q} = 0 \end{cases}$$

where $\alpha = \min\{N_{i,q}, m_i\}$.

Lemma 6. *Under POMLP* $\mathcal{S}^O_{i,q}(\lambda) \le \mathcal{F}^O(N^\lambda_{i,q})$ *where*

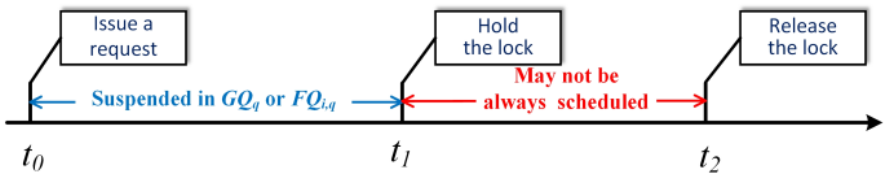
$$\mathcal{F}^O(N^\lambda_{i,q}) = \sum_{j \ne i} \min\{m_i \eta^q_{i,j}, (N_{i,q} + (m_i - 1)N^\lambda_{i,q})m_j\}L_{j,q}$$

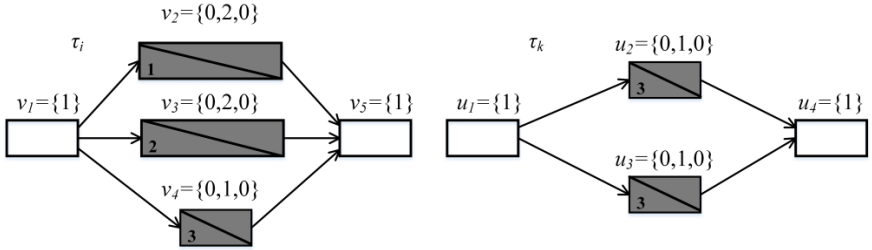


POMIP

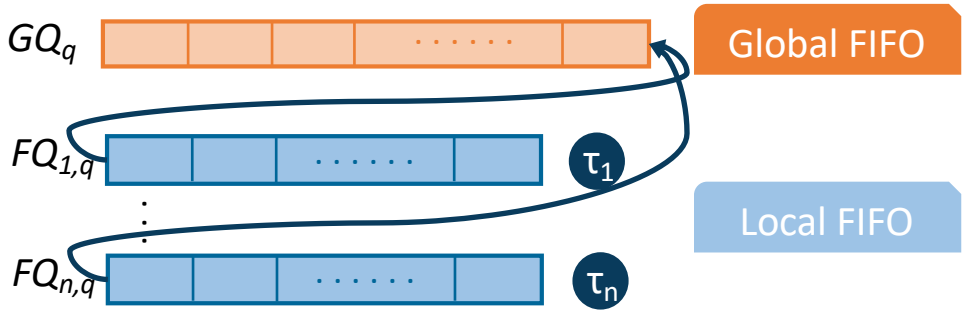
Rules

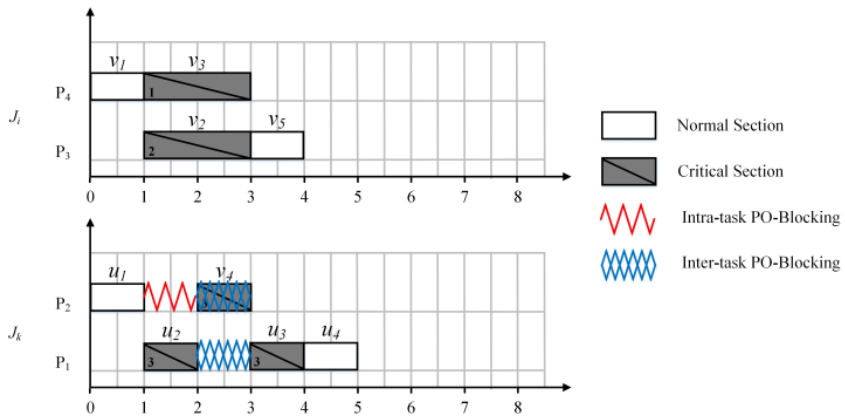
- **Rule 1:** A vertex holding a lock can preempt a scheduled vertex not holding a lock whereas a scheduled vertex holding a lock can not be preempted.
- **Rule 2:** Whenever a lock-holding vertex v is not scheduled and there exists a vertex u (waiting for v to release a lock) such that v is able to be scheduled in u's cluster, v migrates to u's cluster for execution. After releasing the lock, v migrates back to its assigned cluster





Queue structure





POMIP

PO-Blocking Time

$$\mathcal{S}_i(\lambda) = \sum_{\ell_q \in \Theta_i} \left(\mathcal{S}^I_{i,q}(\lambda) + \mathcal{S}^O_{i,q}(\lambda)\right)$$

Intra-task Inter-task

Lemma 8. *Under POMIP* $\mathcal{S}^I_{i,q}(\lambda) \le \mathcal{F}^I(N^\lambda_{i,q})$ *where*

$$\mathcal{F}^I(N^\lambda_{i,q}) = \begin{cases} (N_{i,q} - N^\lambda_{i,q})(m_i - 1)L_{i,q}, & N^\lambda_{i,q} > 0 \\ 0, & N^\lambda_{i,q} = 0 \end{cases}$$

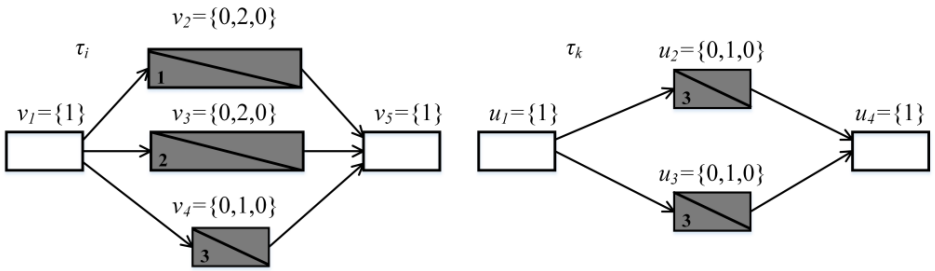
Lemma 10. *Under POMIP*

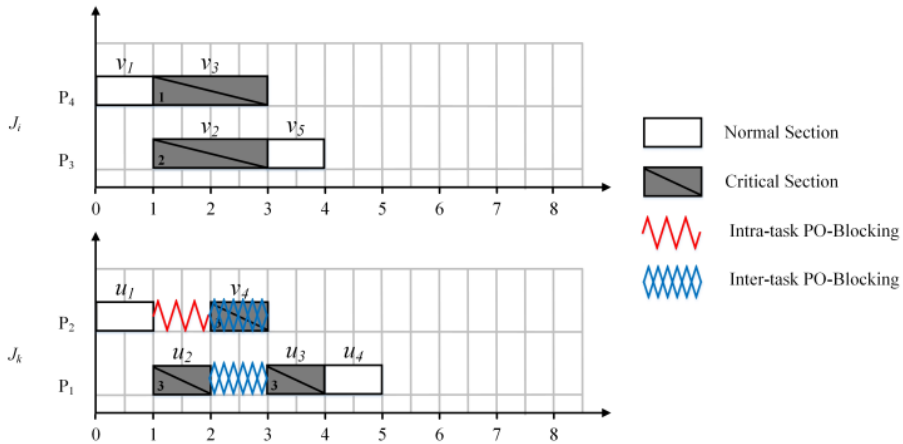
$$\sum_{\ell_q \in \Theta_i} \mathcal{S}^O_{i,q}(\lambda) \le \sum_{\ell_q \in \Theta_i} \mathcal{F}^O(N^\lambda_{i,q})$$

and

$$\mathcal{F}^O(N^\lambda_{i,q}) = \sum_{j \ne i} \left((\eta^q_{i,j} + (m_i - 1)\Delta^q_{i,j})L_{j,q}\right)$$

where $\Delta^q_{i,j} = \min\{N^\lambda_{i,q}, 1\} \min\{\eta^q_{i,j}, N_{i,q}\}$.





4

PART FOUR

SCHEDULABILITY

Schedulability

$$\mathcal{R}_i \leq \max_{\lambda \in \tau_i} \left\{ \frac{\mathcal{C}_i + (m_i - 1) len(\lambda) + \mathcal{S}_i(\lambda)}{m_i} \right\}$$

$$\max_{\lambda \in \tau_i} \left((m_i - 1) len(\lambda) + \mathcal{S}_i(\lambda) \right) \leq (m_i - 1)\mathcal{L}_i + \max_{\lambda \in \tau_i} \{\mathcal{S}_i(\lambda)\}$$

$$\mathcal{R}_i \leq \frac{\mathcal{C}_i + (m_i - 1)\mathcal{L}_i + \max_{\lambda \in \tau_i} \{\mathcal{S}_i(\lambda)\}}{m_i}$$

POMLP/POMIP

$$\mathcal{R}_i \leq \frac{\mathcal{C}_i + (m_i - 1)\mathcal{L}_i + \sum_{\ell_q \in \Theta_i} \max_{N^{\lambda}_{i,q} \in [0, N_{i,q}]} \{\mathcal{F}^I(N^{\lambda}_{i,q}) + \mathcal{F}^O(N^{\lambda}_{i,q})\}}{m_i}$$

Algorithm 1: Processor allocation algorithm.

```
For each τ_i: initialize m_i to be ⌈(C_i − L_i)/(D_i − L_i)⌉ ;
while success ≠ 1 do
    success = 1 ;
    for (each task τ_i) do
        Finding maximum value of corresponding S_i(λ)
          (of the chosen protocol) by enumerating each
          value N^λ_{i,q} ∈ [0, N_{i,q}] for each ℓ_q;
        Compute R_i by (6);
        if (R_i > D_i) then
            m_i ← m_i + 1; success = 0;
    if (Σ_{τ_i ∈ T} m_i > m) then
        return unschedulable;
    if success = 1 then
        return schedulable;
```

5

PART FIVE

EVALUATIONS

Experimental Set-up

Task Graph

Based on Erdös Rényi method $G(|V_i|,p)$

p	0.1
|Vi|	[200,400]
WCET	[300,1500]

Deadline and Period

After $\mathcal{L}_i$ is fixed, D_i is generated according to a ratio between $\mathcal{L}_i$ and D_i randomly chosen in [0.125,0.25]. The period T_i is set to be equal to D_i.

Resource

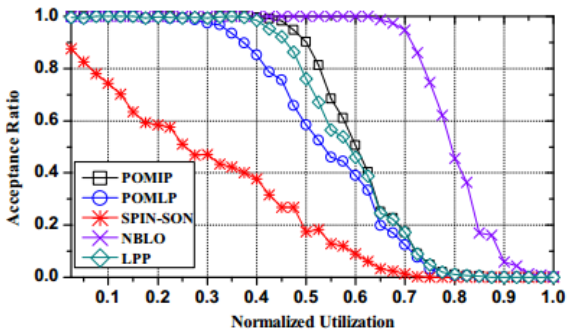
resource types n_r	[1,12]
$\sum_{\tau_i \in \mathcal{T}} N_{i,q}$	[20,440]
$\max_{\forall \tau_i}\{L_{i,q}\}$	[5,100]
$L_{i,q}$	$[5, \max_{\forall \tau_i}\{L_{i,q}\}]$

Basic Configuration

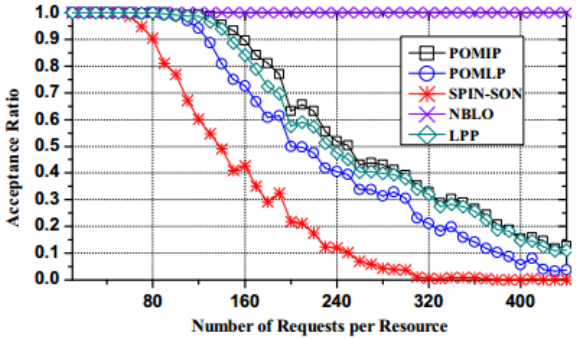
task number n	3
Normalized utilization U_{norm}	0.6
n_r	6
$\sum_{\tau_i \in \mathcal{T}} N_{i,q}$	250
$\max_{\forall \tau_i}\{L_{i,q}\}$	50

Experiment Results

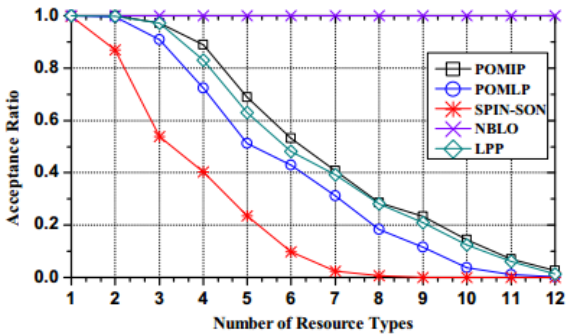
POMIP>LPP>POMLP>SPIN-SON



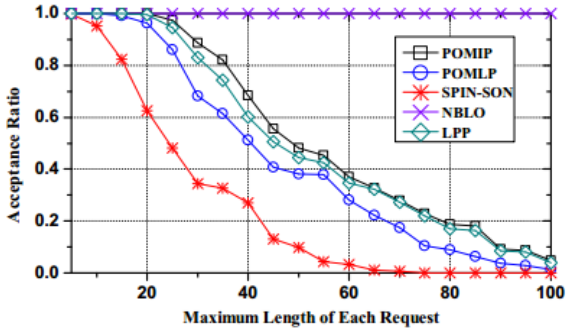
(a) Under different U_{norm}.



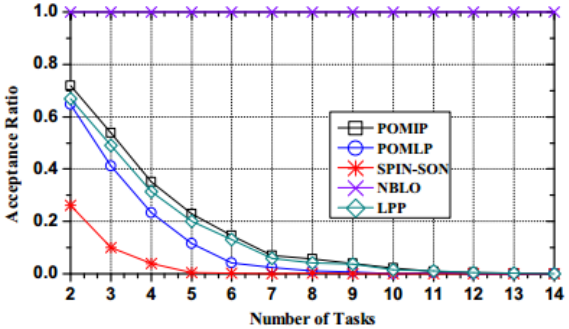
(b) Under different $\sum_{\tau_i \in \mathcal{T}} N_{i,q}$.



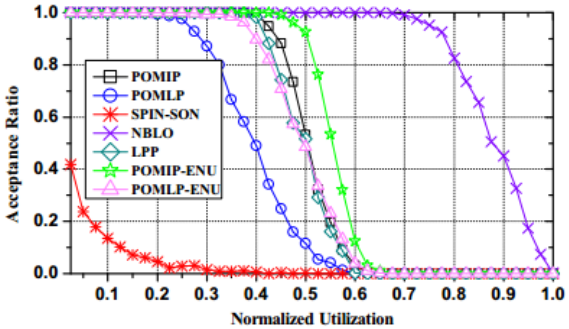
(c) Under different n_r.



(d) Under different $\max_{\forall \tau_i}\{L_{i,q}\}$.



(e) Under different n.



(f) Comparison with POMIP-ENU and POMLP-ENU.

THANKS!