Suspension-Based Locking Protocols for Parallel Real-Time Tasks

Xu Jiang, Nan Guan, Yue Tang, Weichen Liu, Hancong Duan
01 / INTRODUCTION
02 / PRELIMINARY
03 / BLOCKING ANALYSIS
04 / SCHEDULABILITY
05 / EVALUATIONS
INTRODUCTION
Background

**Locking protocols**

- **sequential real-time tasks**
  - well-studied in classical real-time scheduling theory.

- **parallel real-time tasks**
  - still a vastly open problem

**multi-cores processors**
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.

**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.
PART TWO

PRELIMINARY
SYSTEM MODEL

Task Model

- task set: \( \mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\} \)
- platform: \( m \) processors
- each task \( \tau_i: \ G_i = (V_i, E_i) \)
- constrained deadline: \( D_i \leq T_i \)
- each vertex: \( \nu = \{c_{i,1}, c'_{i,1}, \ldots, c'_{i,s-1}, c_{i,s}\} \)

Resource Model

\( \Theta = \{\ell_1, \ldots, \ell_{n_r}\} \)

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 $v_j$.

$$m_i \times (f(J_i) - r(J_i)) = W_i(\lambda) + S_i(\lambda) + I_i(\lambda)$$

- **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** $S_i(\lambda)$: the cumulative length of time on $m_i$ processors being idle when $\lambda$ is suspended.

- **PO-Idle Time** $I_i(\lambda)$: the cumulative length of time on $m_i$ processors being idle when $\lambda$ is not suspended.

$$R_i \leq \max_{\lambda \in \tau_i} \left\{ \frac{C_i + (m_i - 1)\text{len}(\lambda) + S_i(\lambda)}{m_i} \right\}$$
BLOCKING ANALYSIS
**Rule 1:** A vertex holding a lock can preempt a scheduled vertex not holding a lock whereas a scheduled vertex holding a lock cannot 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**

- $FQ_q$: Global FIFO
- $SQ_1$ to $SQ_n$: Local FIFO
Lemma 5. Under POMLP $S_{i,q}^{I}(\lambda) \leq \mathcal{F}^{I}(N_{i,q}^{\lambda})$ where

$$\mathcal{F}^{I}(N_{i,q}^{\lambda}) = \begin{cases} (N_{i,q} - N_{i,q}^{\lambda})(m_{i} - 1)L_{i,q}, & N_{i,q}^{\lambda} > 0 \\ \frac{(\alpha(\alpha - 1)}{2} + (m_{i} - 1)(N_{i,q} - \alpha))L_{i,q}, & N_{i,q}^{\lambda} = 0 \end{cases}$$

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

Lemma 6. Under POMLP $S_{i,q}^{O}(\lambda) \leq \mathcal{F}^{O}(N_{i,q}^{\lambda})$ where

$$\mathcal{F}^{O}(N_{i,q}^{\lambda}) = \sum_{j \neq i} \min\{m_{i}n_{i,j}^{q}, (N_{i,q} + (m_{i} - 1)N_{i,q}^{\lambda})m_{j}\}L_{j,q}$$
**Rules**

- **Rule 1:** A vertex holding a lock can preempt a scheduled vertex not holding a lock whereas a scheduled vertex holding a lock cannot 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**

- $GQ_q$: Global FIFO
- $FQ_{1,q}$, $FQ_{n,q}$: Local FIFO

**Diagram:**
- $GQ_q$: Global FIFO
- $FQ_{1,q}$, $FQ_{n,q}$: Local FIFO
Lemma 8. Under POMIP, $S_{i,q}^I(\lambda) \leq F^I(N_{i,q}^\lambda)$ where

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

Lemma 10. Under POMIP:

$$\sum_{\ell_q \in \Theta_i} S_{i,q}^O(\lambda) \leq \sum_{\ell_q \in \Theta_i} F^O(N_{i,q}^\lambda)$$

and

$$F^O(N_{i,q}^\lambda) = \sum_{j \neq i} \left( (n_{i,j}^q + (m_i - 1)\Delta_{i,j}^q)L_{j,q} \right)$$

where $\Delta_{i,j}^q = \min\{N_{i,q}^\lambda, 1\} \min\{n_{i,j}^q, N_{i,q}\}$. 

\[ P_1 \quad P_2 \quad P_3 \]

\[ t_1 \quad \tau_i \quad \tau_k \]

\[ v_1 = \{1\} \quad v_2 = \{0, 2, 0\} \quad v_3 = \{1\} \quad u_1 = \{1\} \quad u_2 = \{0, 1, 0\} \quad u_3 = \{1\} \]

\[ \text{Intra-task PO-Blocking} \quad \text{Critical Section} \quad \text{Inter-task PO-Blocking} \]
PART FOUR

SCHEDULABILITY
Schedulability

\[ R_i \leq \max_{\lambda \in \tau_i} \left\{ \frac{C_i + (m_i - 1)len(\lambda) + S_i(\lambda)}{m_i} \right\} \]

\[ R_i \leq \frac{C_i + (m_i - 1)L_i + \max_{\lambda \in \tau_i} \{ S_i(\lambda) \}}{m_i} \]

POMLP/POMIP

Algorithm 1: Processor allocation algorithm.

1. For each \( \tau_i \): initialize \( m_i \) to be \( \left\lfloor \frac{C_i - L_i}{D_i - L_i} \right\rfloor \);
2. while success \( \neq 1 \) do
   3. \( success = 1 \);
   4. for (each task \( \tau_j \)) do
      5. Finding maximum value of corresponding \( S_i(\lambda) \) (of the chosen protocol) by enumerating each value \( N_{i,q}^\lambda \in [0, N_{i,q}] \) for each \( \ell_q \);
      6. Compute \( R_i \) by (6);
      7. if \( (R_i > D_i) \) then
         8. \( m_i \leftarrow m_i + 1; success = 0; \)
         9. if \( (\sum_{\tau \in \mathcal{T}} m_i > m) \) then
            10. return unschedulable;
      11. if success = 1 then
         12. return schedulable.
Experimental Set-up

Task Graph

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

Deadline and Period

After $L_i$ is fixed, $D_i$ is generated according to a ratio between $L_i$ and $D_i$ randomly chosen in $[0.125, 0.25]$. The period $T_i$ is set to be equal to $D_i$.

Resource

<table>
<thead>
<tr>
<th>resource types $n_r$</th>
<th>$[1, 12]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_{\tau \in T} N_{i,q}$</td>
<td>$[20, 440]$</td>
</tr>
<tr>
<td>$\max_{\tau \in T} {L_{i,q}}$</td>
<td>$[5, 100]$</td>
</tr>
<tr>
<td>$L_{i,q}$</td>
<td>$[5, \max_{\tau \in T} {L_{i,q}}]$</td>
</tr>
</tbody>
</table>

Basic Configuration

<table>
<thead>
<tr>
<th>task number $n$</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized utilization $U_{norm}$</td>
<td>0.6</td>
</tr>
<tr>
<td>$n_r$</td>
<td>6</td>
</tr>
<tr>
<td>$\sum_{\tau \in T} N_{i,q}$</td>
<td>250</td>
</tr>
<tr>
<td>$\max_{\tau \in T} {L_{i,q}}$</td>
<td>50</td>
</tr>
</tbody>
</table>
Experiment Results

POMIP > LPP > POMLP > SPIN-SON

(a) Under different $U_{norm}$.

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

(c) Under different $n_T$.

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

(e) Under different $n$.

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