An Asymptotically Optimal Real-Time Locking Protocol for Clustered Scheduling under Suspension-Aware Analysis

12/5/2012
RTSS’12 WIP

Björn B. Brandenburg
bbb@mpi-sws.org
### Background: Suspension-Based Multiprocessor Real-Time Locking

<table>
<thead>
<tr>
<th>Blocking Optimality</th>
<th>suspension oblivious</th>
<th>suspension aware</th>
</tr>
</thead>
<tbody>
<tr>
<td>[— &amp; Anderson, 2010]</td>
<td>CPU demand is over-approximated</td>
<td>CPU demand is modeled accurately</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>How are suspensions analyzed?</th>
<th>Lower bound on maximum priority inversion blocking $\max_i{B_i}$</th>
<th>$\Omega(m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m = #CPUs$</td>
<td>$\Omega(n)$</td>
</tr>
<tr>
<td></td>
<td>$n = #tasks$</td>
<td></td>
</tr>
</tbody>
</table>

# Asymptotically Optimal Locking Protocols

<table>
<thead>
<tr>
<th>JLFP</th>
<th>Suspension Oblivious</th>
<th>Suspension Aware</th>
</tr>
</thead>
<tbody>
<tr>
<td>job-level fixed-priority</td>
<td>Any JLFP Scheduler</td>
<td>EDF w/ Implicit Deadlines</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Suspension Aware</th>
<th>Any JLFP Scheduler</th>
</tr>
</thead>
</table>

## Partitioned
- (no migrations)

## Global
- (jobs migrate freely)

## Clustered
- (jobs migrate only among subset of processors)

# Asymptotically Optimal Locking Protocols

<table>
<thead>
<tr>
<th>JLFP (job-level fixed-priority)</th>
<th>Suspension Oblivious Any JLFP Scheduler</th>
<th>Suspension Aware EDF w/ Implicit Deadlines</th>
<th>Suspension Aware Any JLFP Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Partitioned</strong> (no migrations)</td>
<td>P-OMLP ✅ Devices [— &amp; Anderson, 2010]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Global</strong> (jobs migrate freely)</td>
<td>G-OMLP ✅ Devices [— &amp; Anderson, 2010]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Clustered</strong> (jobs migrate only among subset of processors)</td>
<td>C-OMLP ✅ Devices [— &amp; Anderson, 2011]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[— & Anderson, 2010]; Optimality Results for Multiprocessor Real-Time Locking, RTSS 2010.  
## Asymptotically Optimal Locking Protocols

<table>
<thead>
<tr>
<th>JLFP = job-level fixed-priority</th>
<th>Suspension Oblivious Any JLFP Scheduler</th>
<th>Suspension Aware EDF w/ Implicit Deadlines</th>
<th>Suspension Aware Any JLFP Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Partitioned</strong> (no migrations)</td>
<td><strong>P-OMLP</strong> ✔️ [— &amp; Anderson, 2010]</td>
<td><strong>SPFP</strong> (asymptotical tightness) [— &amp; Anderson, 2010]</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Clustered</strong> (jobs migrate only among subset of processors)</td>
<td><strong>C-OMLP</strong> ✔️ [— &amp; Anderson, 2011]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Asymptotically Optimal Locking Protocols

<table>
<thead>
<tr>
<th>JLFP</th>
<th>Suspension Oblivious</th>
<th>Suspension Aware</th>
<th>Suspension Aware</th>
</tr>
</thead>
<tbody>
<tr>
<td>job-level fixed-priority</td>
<td>Any JLFP Scheduler</td>
<td>EDF w/ Implicit Deadlines</td>
<td>Any JLFP Scheduler</td>
</tr>
</tbody>
</table>

## Partitioned

- **P-OMLP**
  - [Block et al., 2010]
- **SPFP** (asymptotical tightness)
  - [— & Anderson, 2010]
- **FMLP** (practical protocol)
  - [—, 2011]

## Global

- **G-OMLP**
  - [— & Anderson, 2010]
- **FMLP**
  - [Block et al., 2007]

## Clustered

- **C-OMLP**
  - [— & Anderson, 2011]

---

# Asymptotically Optimal Locking Protocols

<table>
<thead>
<tr>
<th>JLFP = job-level fixed-priority</th>
<th><strong>Suspension Oblivious</strong> Any JLFP Scheduler</th>
<th><strong>Suspension Aware</strong> Any JLFP Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Partitioned</strong> (no migrations)</td>
<td>P-OMLP [— &amp; Anderson, 2010]</td>
<td></td>
</tr>
<tr>
<td><strong>Global</strong> (jobs migrate freely)</td>
<td>G-OMLP [— &amp; Anderson, 2010]</td>
<td></td>
</tr>
<tr>
<td><strong>Clustered</strong> (jobs migrate only among subset of processors)</td>
<td>C-OMLP [— &amp; Anderson, 2011]</td>
<td></td>
</tr>
</tbody>
</table>

This Work

The Generalized FMLP+

---

The Generalized FMLP$^+$

**FIFO Multiprocessor Locking Protocol**

The Goal

$\max_i \{B_i\} = O(n)$ maximum priority inversion blocking

- $n = \#tasks$

- Use a FIFO queue!

The Problem [—, 2011]

- Priority **inheritance** leads to $\Omega(\phi)$ blocking.
  - $\phi = \text{ratio largest to shortest period, unbounded in general}$

- Priority **boosting** also leads to $\Omega(\phi)$ blocking…

**New Solution:** FIFO Boosting

prioritize by order of lock request & release times

- cannot preempt
- may preempt *(one per task)*
- may not preempt

**critical sections**

- $CS_b$
- $CS_x$
- $CS_a$

**analyzed job**

- $J_i$

released

$\Rightarrow O(n)$ preemptions
Thanks!

I’ll be happy to answer your questions on Friday…