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

12/5/2012
RTSS'12 WIP

Max
Planck
Institute
for
Software Systems

Björn B. Brandenburg
bbb@mpi-sws.org

Background: Suspension-Based Multiprocessor Real-Time Locking

Blocking Optimality [– & Anderson, 2010]	suspension **oblivious**	suspension **aware**
How are suspensions analyzed?	CPU demand is **over-approximated**	CPU demand is modeled **accurately**
Lower bound on maximum priority inversion blocking $max_i\{B_i\}$	$\Omega(m)$ m = #CPUs	$\Omega(n)$ n = #tasks

[— & Anderson, 2010] *Optimality Results for Multiprocessor Real-Time Locking*, RTSS 2010.

Asymptotically Optimal Locking Protocols

JLFP = job-level fixed-priority	Suspension Oblivious Any JLFP Scheduler	Suspension Aware EDF w/ Implicit Deadlines	Suspension Aware Any JLFP Scheduler
Partitioned (no migrations)			
Global (jobs migrate freely)			
Clustered (jobs migrate only among subset of processors)			

[Block et al., 2007] *A Flexible Real-Time Locking Protocol for Multiprocessors*, RTCSA 2007.

[— & Anderson, 2010] *Optimality Results for Multiprocessor Real-Time Locking*, RTSS 2010.

[— & Anderson, 2011] *Real-Time Resource-Sharing under Clustered Scheduling: Mutex, Reader-Writer, and k-Exclusion Locks*, EMSOFT 2011.

[—, 2011] *Scheduling and Locking in Multiprocessor Real-Time Operating Systems*, PhD thesis, UNC, 2011.

Asymptotically Optimal Locking Protocols

JLFP = job-level fixed-priority	Suspension Oblivious Any JLFP Scheduler	Suspension Aware EDF w/ Implicit Deadlines	Suspension Aware Any JLFP Scheduler
Partitioned (no migrations)	**P-OMLP** ✓ [— & Anderson, 2010]		
Global (jobs migrate freely)	**G-OMLP** ✓ [— & Anderson, 2010]		
Clustered (jobs migrate only among subset of processors)	**C-OMLP** ✓ [— & Anderson, 2011]		

[Block et al., 2007] *A Flexible Real-Time Locking Protocol for Multiprocessors*, RTCSA 2007.

[— & Anderson, 2010] *Optimality Results for Multiprocessor Real-Time Locking*, RTSS 2010.

[— & Anderson, 2011] *Real-Time Resource-Sharing under Clustered Scheduling: Mutex, Reader-Writer, and k-Exclusion Locks*, EMSOFT 2011.

[—, 2011] *Scheduling and Locking in Multiprocessor Real-Time Operating Systems*, PhD thesis, UNC, 2011.

Asymptotically Optimal Locking Protocols

JLFP = job-level fixed-priority	Suspension Oblivious Any JLFP Scheduler	Suspension Aware EDF w/ Implicit Deadlines	Suspension Aware Any JLFP Scheduler
Partitioned (no migrations)	**P-OMLP** ✓ [— & Anderson, 2010]	**SPFP** (asymptotical tightness) [— & Anderson, 2010] **FMLP+** (practical protocol) [—, 2011] ✓	
Global (jobs migrate freely)	**G-OMLP** ✓ [— & Anderson, 2010]		
Clustered (jobs migrate only among subset of processors)	**C-OMLP** ✓ [— & Anderson, 2011]		

[Block et al., 2007] *A Flexible Real-Time Locking Protocol for Multiprocessors*, RTCSA 2007.

[— & Anderson, 2010] *Optimality Results for Multiprocessor Real-Time Locking*, RTSS 2010.

[— & Anderson, 2011] *Real-Time Resource-Sharing under Clustered Scheduling: Mutex, Reader-Writer, and k-Exclusion Locks*, EMSOFT 2011.

[—, 2011] *Scheduling and Locking in Multiprocessor Real-Time Operating Systems*, PhD thesis, UNC, 2011.

Asymptotically Optimal Locking Protocols

JLFP = job-level fixed-priority	Suspension Oblivious Any JLFP Scheduler	Suspension Aware EDF w/ Implicit Deadlines	Suspension Aware Any JLFP Scheduler
Partitioned (no migrations)	**P-OMLP** ✓ [— & Anderson, 2010]	**SPFP** (asymptotical tightness) [— & Anderson, 2010] **FMLP+** (practical protocol) [—, 2011] ✓	
Global (jobs migrate freely)	**G-OMLP** ✓ [— & Anderson, 2010]	**FMLP** ! [Block et al., 2007]	?
Clustered (jobs migrate only among subset of processors)	**C-OMLP** ✓ [— & Anderson, 2011]	?	?

[Block et al., 2007] *A Flexible Real-Time Locking Protocol for Multiprocessors*, RTCSA 2007.
[— & Anderson, 2010] *Optimality Results for Multiprocessor Real-Time Locking*, RTSS 2010.
[— & Anderson, 2011] *Real-Time Resource-Sharing under Clustered Scheduling: Mutex, Reader-Writer, and k-Exclusion Locks*, EMSOFT 2011.
[—, 2011] *Scheduling and Locking in Multiprocessor Real-Time Operating Systems*, PhD thesis, UNC, 2011.

Asymptotically Optimal Locking Protocols

JLFP = job-level fixed-priority	Suspension Oblivious Any JLFP Scheduler	Suspension Aware Any JLFP Scheduler
Partitioned (no migrations)	**P-OMLP** ✓ [— & Anderson, 2010]	**This Work** The Generalized FMLP+
Global (jobs migrate freely)	**G-OMLP** ✓ [— & Anderson, 2010]	✓
Clustered (jobs migrate only among subset of processors)	**C-OMLP** ✓ [— & Anderson, 2011]	

[Block et al., 2007] *A Flexible Real-Time Locking Protocol for Multiprocessors*, RTCSA 2007.

[— & Anderson, 2010] *Optimality Results for Multiprocessor Real-Time Locking*, RTSS 2010.

[— & Anderson, 2011] *Real-Time Resource-Sharing under Clustered Scheduling: Mutex, Reader-Writer, and k-Exclusion Locks*, EMSOFT 2011.

[—, 2011] *Scheduling and Locking in Multiprocessor Real-Time Operating Systems*, PhD thesis, UNC, 2011.

The Generalized FMLP+

FIFO Multiprocessor Locking Protocol

The Goal

- ***max*$_i${B_i} = $O(n)$** maximum priority inversion blocking
  - *n = #tasks*
- Use a **F**IFO queue!

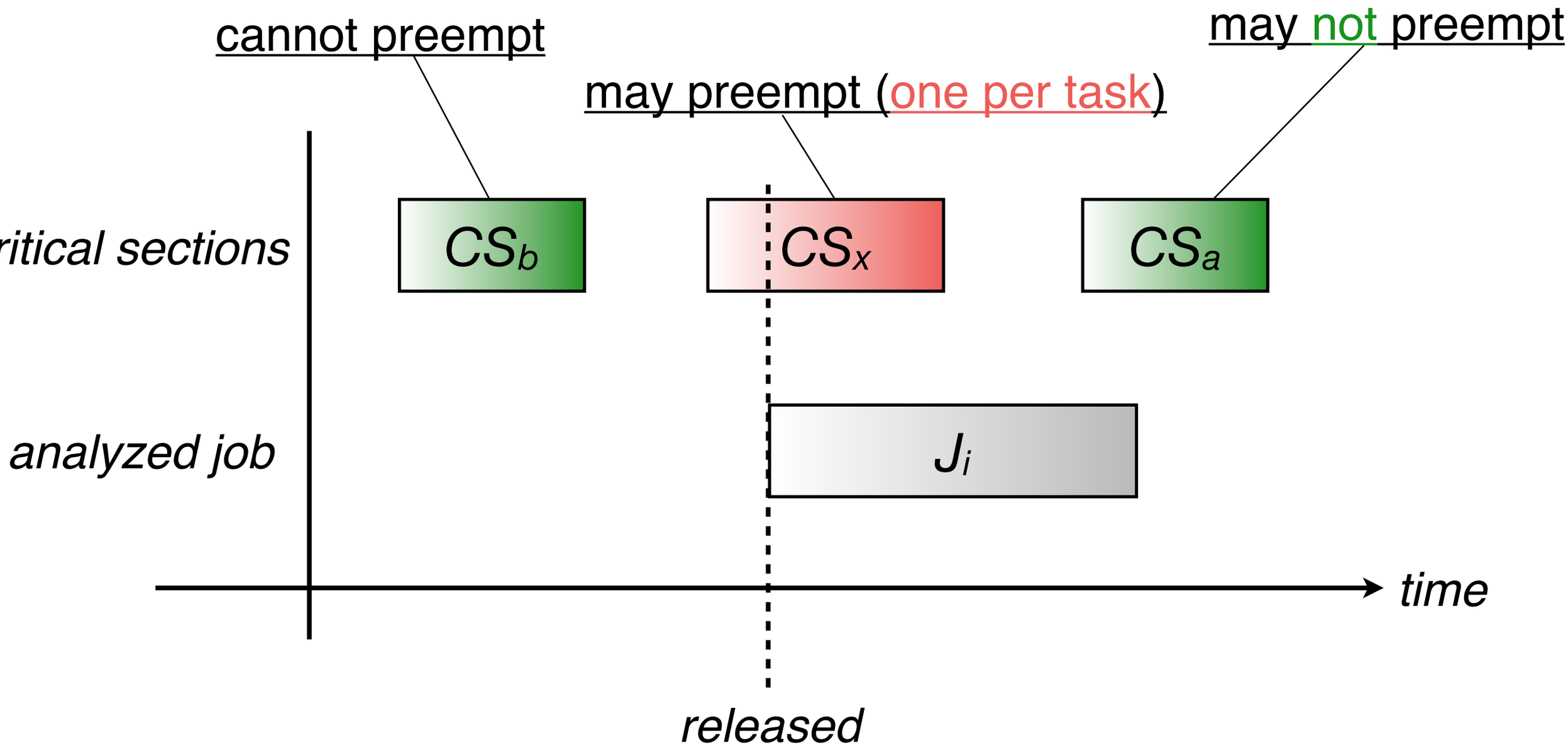
The Problem [—, 2011]

- Priority **inheritance** leads to $\Omega(\phi)$ blocking.
  - ϕ = ratio largest to shortest period, **unbounded** in general
- Priority **boosting** also leads to $\Omega(\phi)$ blocking…

[—, 2011] *Scheduling and Locking in Multiprocessor Real-Time Operating Systems*, PhD thesis, UNC, 2011.

New Solution: FIFO Boosting

prioritize by order of lock request & release times



➡ ***O(n)*** preemptions

Thanks!

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