Orthogonal to Multiprocessor Resource Sharing Protocols: How to Share?

Jian-Jia Chen

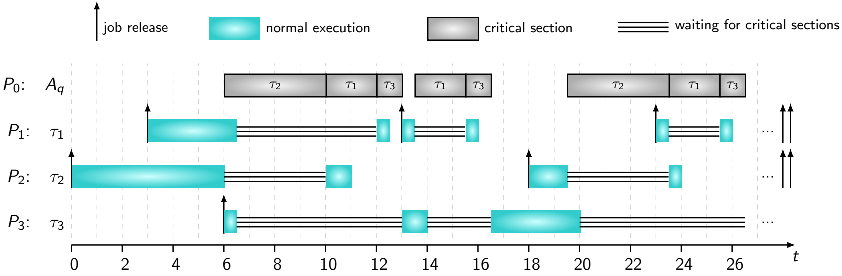
TU Dortmund, Germany

27,06,2017 at RTSOPS

technische universität dortmund

SFB 876 Verfügbarkeit von Information durch Analyse unter Ressourcenbeschränkung

An Example of Locking Protocols



- Distributed PCP in the above example
- Semaphores in multiprocessor systems: remote blocking due to mutual exclusion (that may cause self-suspension or task spinning)

Two Correlated Problems

- Scheduler Design Problem
  - *Design locking protocols* to synchronize the critical sections
  - *Design scheduling policies* to schedule the synchronized tasks
  - *Partition the tasks* to processors if the protocol is restricted to partitioned or semi-partitioned scheduling
- Schedulability Test Problem
  - *Validate* the schedulability of a scheduling algorithm.

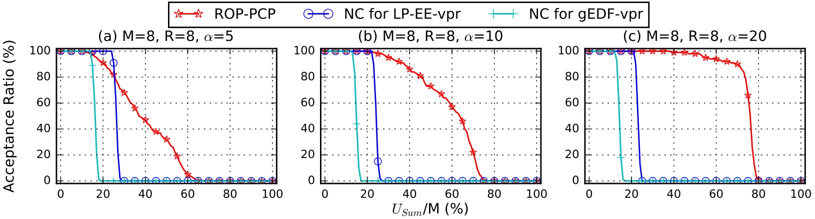
Existing Protocols

- Partitioned scheduling
    - MPCP (extension of PCP to multiprocessor systems), 1990
    - MSRP (extension of SRP to multiprocessor systems), 2001
    - MrsP (combination of spin-based and suspension-based protocols), 2013
- Semi-partitioned scheduling
    - DPCP (synchronization processors for critical sections), 1988
- Global scheduling
    - FMLP, 2007
    - FMLP$^+$, 2014
    - DFLP, 2014
    - gEDF-vpr (bounded speedup factor), 2014
    - etc.

Results with Bounded Speedup Factors

- We have to consider all the following factors together
  - task partitioning or priority assignment in global scheduling,
  - the locking protocols, and
  - the schedulability tests
- ROP-PCP by Huang et al. RTSS 2016
  - speedup factor of 11-6/M
- LP-EE-vpr by Andersson and Raravi RTS 2014
  - speedup factor of 8 (when $R \leq M$)
  - M virtual processors at speed $\frac{1}{2}$ for the non-critical sections
  - M virtual processors at speed $\frac{1}{2}$ for the critical sections
  - each segment assigned with a fixed relative deadline

Simulation Results



- Necessary condition (NC) for LP-EE-vpr (by Andersson and Raravi RTS 2014) and gEDF-vpr (by Andersson and Easwaran RTS 2010)
- Sufficient condition for ROP-PCP (by Huang et al. RTSS 2016)
- Configurations
  - M: number of processors
  - α: control variable for the critical section lengths (a smaller α leads to shorter critical sections)
  - R: number of semaphores

Blocking Time

- Each task may necessarily suffer from one direct blocking
- Other types of blocking time should be avoided

Protocol	Critical factor	Remark
(M)PIP	N	Chained blocking
MPCP	N	partitioning
MrsP	M	FIFO
gEDF-vpr	R	Virtualization
Ideal	**1**	**Largest**

N:number of tasks, M: number of processors, R: number of shared resources

Existing Partitioning Algorithms

- synchronization-aware partitioning (Lakshmanan et al. RTSS 2009), called SPA
  - put the tasks that share resources into the same *macrotask*
  - try to put a macrotask to one processor
  - if not, split into multiple processors
- block-aware partitioning (Nemati et al. OPODIS 2010), called BPA
  - the remote blocking time is used to define the weight of a task
- greedy slacker (Wieder and Brandenburg, SIES 2013)
  - a simple strategy explained next

Greedy Slacker (GS)

<u>embarrassingly simple:</u>

- disregard graph structure
- greedily try to maximize minimum slack

for each task T_i in order of increasing period:
  for each processor C_k:
    compute slack when T_i assigned to C_k
  if there is no C_r such that minimum slack $\geq$ 0:
    fail
  else:
    assign T_i to C_r s.t. minimum slack is maximized

from Wieder and Brandenburg

Greedy Slacker (GS)

embarrassingly simple:

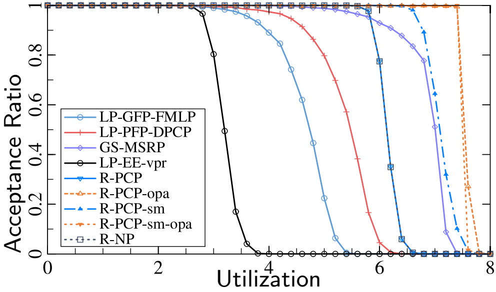
- disregard graph structure
- greedily try to maximize minimum slack

for each task T_i in order of increasing period:
  for each processor C_k:
    compute slack when T_i assigned to C_k
  if there is no C_r such that minimum slack $\geq$ 0:
    fail
  else:
    assign T_i to C_r s.t. minimum slack is maximized

from Wieder and Brandenburg

Surprisingly, both SPA and BPA led to significantly lower schedulability than GS. This effect was unexpected since both the SPA and BPA were particularly designed for scenarios with resource sharing, while GS is resource-oblivious.

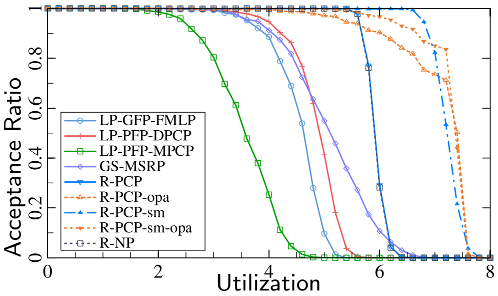
Another Evaluation Result



- Setup
  - 8 cores
  - One critical section per task
- Existing methods
  - LP-GFP-FMLP[4], LP-PFP-MPCP[1], LP-PFP-DPCP[1]
  - LL-EE-vpr[3]
  - GS-MSRP[2]
- Our Methods
  - R-PCP, R-NP
  - R-PCP-opa, R-PCP-sm, R-PCP-sm-opa

1. B. B. Brandenburg. Improved analysis and evaluation of realtime semaphore protocols for P-FP scheduling. In Real-Time and mbedded Technology and Applications Symposium, RTAS, pages 141-152, 2013.
2. A. Wieder and B. B. Brandenburg. On spin locks in AUTOSAR: blocking analysis of fifo, unordered, and priority-ordered spin locks. In Real-Time Systems Symposium, pages 45-56, 2013.
3. B. Andersson and G. Raravi. Real-time scheduling with resource sharing on heterogeneous multiprocessors. Real-Time Systems, 50(2):270-314, 2014.
4. M. Yang, A. Wieder, and B. B. Brandenburg. Global real-time semaphore protocols: A survey, unified analysis, and comparison. In Real-Time Systems Symposium (RTSS), pages 1-12, 2015

Another Evaluation Result



- Setup
  - 8 cores
  - Up to 5 critical sections per task
- Existing methods
  - LP-GFP-FMLP[4], LP-PFP-MPCP[1], LP-PFP-DPCP[1]
  - LL-EE-vpr[3]
  - GS-MSRP[2]
- Our Methods
  - R-PCP, R-NP
  - R-PCP-opa, R-PCP-sm, R-PCP-sm-opa

1. B. B. Brandenburg. Improved analysis and evaluation of realtime semaphore protocols for P-FP scheduling. In Real-Time and mbedded Technology and Applications Symposium, RTAS, pages 141-152, 2013.
2. A. Wieder and B. B. Brandenburg. On spin locks in AUTOSAR: blocking analysis of fifo, unordered, and priority-ordered spin locks. In Real-Time Systems Symposium, pages 45-56, 2013.
3. B. Andersson and G. Raravi. Real-time scheduling with resource sharing on heterogeneous multiprocessors. Real-Time Systems, 50(2):270-314, 2014.
4. M. Yang, A. Wieder, and B. B. Brandenburg. Global real-time semaphore protocols: A survey, unified analysis, and comparison. In Real-Time Systems Symposium (RTSS), pages 1-12, 2015

A Closer Look

- LP-PFP-MPCP[1] and LP-PFP-DPCP[1]
  - LP-based schedulability test
  - partition fixed-priority (WFD fitting)
  - MPCP/DPCP protocol
- GS-MSRP
  - pseudo-polynomial-time schedulability test (OPA-compatible)
  - partition fixed-priority (OPA-based packing)
  - MSRP protocol
- R-PCP and R-NP
  - resource oriented partitioning
  - pseudo-polynomial-time schedulability test (suspension-aware)
  - rate-monotonic priority assignment
  - PCP or non-preemptive protocol in the synchronization processor

Open Problems/Questions

- What are we actually comparing
  - partitioning algorithms?
  - schedulability tests?
  - resource synchronization protocols?
- Is the comparison fair?
  - LP-PFP-MPCP and LP-PFP-DPCP do not use the best partitioned algorithm and priority assignments
  - SPA and BPA algorithms are not the best strategies
  - To fairly compare these protocols, partitioning algorithms designed for the MPCP, MSRP, and MrsP are needed.

Open Problems/Questions

- What are we actually comparing
  - partitioning algorithms?
  - schedulability tests?
  - resource synchronization protocols?
- Is the comparison fair?
  - LP-PFP-MPCP and LP-PFP-DPCP do not use the best partitioned algorithm and priority assignments
  - SPA and BPA algorithms are not the best strategies
  - To fairly compare these protocols, partitioning algorithms designed for the MPCP, MSRP, and MrsP are needed.
- Partitioning algorithms elegantly designed for constrained-deadline task systems are needed.

Open Problems/Questions

- What are we actually comparing
  - partitioning algorithms?
  - schedulability tests?
  - resource synchronization protocols?
- Is the comparison fair?
  - LP-PFP-MPCP and LP-PFP-DPCP do not use the best partitioned algorithm and priority assignments
  - SPA and BPA algorithms are not the best strategies
  - To fairly compare these protocols, partitioning algorithms designed for the MPCP, MSRP, and MrsP are needed.
- Partitioning algorithms elegantly designed for constrained-deadline task systems are needed.
- Do the patterns of the critical sections significantly affect the difficulty for designing good partitioning algorithms?

Conclusion

- How much do we know about the above open problems?
  - the same as what John Snow knows?
  - a little bit better than what John Snow knows?
  - really?