Complexity of Multiprocessor Blocking Analysis with Nested Critical Sections

Alexander Wieder
Björn Brandenburg
Max Planck Institute for Software Systems

RTSS 2014
20141203
This talk

Complexity of tight blocking analysis for:

- multiprocessor systems
- locks with strong ordering guarantees
- nested critical sections
Nested Locks on Multiprocessors

T_1

lock(RED);
unlack(RED);

T_2

lock(RED);
lock(GREEN);
unlock(GREEN);
unlock(RED);
unlock(RED);

T_3

lock(GREEN);
unlock(GREEN);
Nested Locks on Multiprocessors

Critical section for RED contains nested critical section for GREEN.

CPU 1

lock(RED);
  ...
  ...
  ...
  ...
unlock(RED);

CPU 2

lock(RED);
  lock(GREEN);
  ...
  unlock(GREEN);
  ...
unlock(RED);

CPU 3

lock(GREEN);
  ...
  ...
unlock(GREEN);
Nested Locks on Multiprocessors

CPU₁

T₁

lock(RED); .
unlock(RED);

CPU₂

lock(RED);

T₂

lock(RED);
lock(GREEN);

CPU₃

T₃

lock(GREEN);
unlock(GREEN);

Worst-case blocking duration?
The Blocking Analysis Problem

lock(<RED>);  
.  
unlock(<RED>);  

Worst-case blocking duration?
The Blocking Analysis Problem

Blocking Analysis Problem:

Bound the blocking duration that a task can incur in the worst case.
The Blocking Analysis Problem

Blocking Analysis Problem:

Worst-case blocking duration?

Bound the blocking duration that a task can incur in the worst case.

Tight Blocking Bounds:

There is a schedule in which the blocking bound is reached.

lock(RED);
unlock(RED);
Main Result

**multiprocessor system**

locks with **strong** ordering guarantees

nested critical sections
Main Result

multiprocessor system
locks with strong ordering guarantees
nested critical sections

FIFO/priority ordering
Main Result

**Main Result**

* multiprocessor system
  * locks with strong ordering guarantees
  * nested critical sections

Blocking analysis

NP-hard!
Context: Mutex Locks

protect shared resources such as
• shared bus
• shared data structures
• peripheral devices
ensure mutual exclusion

protect shared resources such as
- shared bus
- shared data structures
- peripheral devices
Context: Mutex Locks

- Ensure mutual exclusion
- Protect shared resources such as:
  - Shared bus
  - Shared data structures
  - Peripheral devices

Mandated by AUTOSAR, ...
multiprocessor system
locks with strong ordering guarantees
nested critical sections

Blocking analysis
NP-hard!

So what?
Formally, most interesting scheduling problems are hard!
Complexity of Scheduling-Related Problems

• **Response-Time Analysis**  (Eisenbrand and Rothvoß, 2008)
• **Deciding Periodic Task Set Feasibility**  (Leung and Whitehead, 1982)
• **Scheduling Task Sets with Self-Suspensions**  (Ridouard et al., 2006)
• [...]
Complexity of Scheduling-Related Problems

- Response-Time Analysis (Eisenbrand and Rothvoß, 2008)
- Deciding Periodic Task Set Feasibility (Leung and Whitehead, 1982)
- Scheduling Task Sets with Self-Suspensions (Ridouard et al., 2006)
- [...]
Complexity of Scheduling-Related Problems

- Response-Time Analysis (Eisenbrand and Rothvoß, 2008)
- Deciding Periodic Task Set Feasibility (Leung and Whitehead, 1982)
- Scheduling Task Sets with Self-Suspensions (Ridouard et al., 2006)
- [...]
# Blocking Analysis Complexity

for commonly used protocols

<table>
<thead>
<tr>
<th>Nested Critical Sections</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Uniprocessor</td>
</tr>
<tr>
<td>no</td>
<td></td>
</tr>
<tr>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
Blocking Analysis Complexity
for commonly used protocols

Architecture

<table>
<thead>
<tr>
<th>Nested Critical Sections</th>
<th>Uniprocessor</th>
<th>Multiprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>polynomial</td>
<td></td>
</tr>
<tr>
<td>yes</td>
<td>polynomial</td>
<td></td>
</tr>
</tbody>
</table>

PCP: find longest critical section
### Blocking Analysis Complexity for commonly used protocols

<table>
<thead>
<tr>
<th>Nested Critical Sections</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>LP-Based Analysis takes polynomial time</td>
</tr>
<tr>
<td>no</td>
<td>Uniprocessor: polynomial</td>
</tr>
</tbody>
</table>

LP-Based Analysis takes polynomial time
# Blocking Analysis Complexity

for commonly used protocols

<table>
<thead>
<tr>
<th>Nested Critical Sections</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>Uniprocessor: polynomial</td>
</tr>
<tr>
<td>yes</td>
<td>Uniprocessor: polynomial</td>
</tr>
</tbody>
</table>
## Blocking Analysis Complexity

for commonly used protocols

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Uniprocessor</th>
<th>Multiprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nested Critical Sections</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>polynomial</td>
<td>polynomial</td>
</tr>
<tr>
<td>yes</td>
<td>polynomial</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>
### Blocking Analysis Complexity
for commonly used protocols

<table>
<thead>
<tr>
<th>Nested Critical Sections</th>
<th>Uniprocessor</th>
<th>Multiprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>polynomial</td>
<td>polynomial</td>
</tr>
<tr>
<td>yes</td>
<td>polynomial</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>

**What makes it difficult on multiprocessors with nesting?**
Outline

• Introduction ✔
• Intuition: Why does nesting make the analysis difficult?
• Reduction: From Multiple Choice Matching to Blocking Analysis
• Summary and Conclusion
What makes the blocking analysis difficult?

- **Architecture**
  - Uniprocessor
    - Nested Critical Sections: no
    - Polynomial time
  - Multiprocessor
    - Nested Critical Sections: yes
    - Polynomial time
    - FIFO-ordered locks
    - NP-hard

- **FIFO-ordered locks:** processor-local problems:
  - Greedy approach to determine worst case
What makes the blocking analysis difficult?

<table>
<thead>
<tr>
<th>Nested Critical Sections</th>
<th>Uniprocessor</th>
<th>Multiprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>polynomial</td>
<td>polynomial</td>
</tr>
<tr>
<td>yes</td>
<td>polynomial</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>

FIFO-ordered locks:

not a processor-local problem: Consider all critical sections from all processors at once!
Non-locality in analysis of nested locks

```
lock(RED);
  .
  .
  .
  .
unlock(RED);

lock(GREEN);
  .
  .
  .
  .
unlock(GREEN);
unlock(RED);
```
Non-locality in analysis of nested locks

lock(RED);
.
unlock(RED);

lock(RED);
.
lock(GREEN);
.
unlock(GREEN);
.
unlock(RED);
.
unlock(GREEN);

CPU₁

CPU₂

CPU₃

T₁

T₂

T₃

Worst-case blocking duration?
Each critical section can be blocked by at most one critical section for the same resource from each remote processor.
Non-locality in analysis of nested locks

One of these critical sections can block.

CPU\(_1\)

\(T_1\)

\[
\text{lock(RED);} \\
\quad \text{.} \\
\quad \text{.} \\
\quad \text{.} \\
\text{unlock(RED);} \\
\]

CPU\(_2\)

\(T_2\)

\[
\text{lock(RED);} \\
\quad \text{.} \\
\quad \text{.} \\
\quad \text{.} \\
\text{lock(GREEN);} \\
\text{unlock(GREEN);} \\
\quad \text{.} \\
\quad \text{.} \\
\text{unlock(RED);} \\
\text{unlock(RED);} \\
\]

CPU\(_3\)

\(T_3\)

\[
\text{lock(GREEN);} \\
\quad \text{.} \\
\quad \text{.} \\
\quad \text{.} \\
\text{unlock(GREEN);} \\
\]
Non-locality in analysis of nested locks

CPU₁  CPU₂  CPU₃

T₁  T₂  T₃

lock(RED);
.
.
unlock(RED);

lock(RED);
lock(GREEN);
.
unlock(GREEN);
unlock(RED);
unlock(RED);

lock(GREEN);
.
.
unlock(GREEN);
Non-locality in analysis of nested locks

$T_1$

lock(RED);
.
.
.
.
.
.
unlock(RED);

$T_2$

lock(RED);
  lock(GREEN);
  unlock(GREEN);
  unlock(RED);
unlock(RED);

$T_3$

lock(GREEN);
.
.
.
unlock(GREEN);
Non-locality in analysis of nested locks

CPU\(_1\)

CPU\(_2\)

CPU\(_3\)

\(T_1\)

```c
lock(RED);
.
.
.
.
.
.
unlock(RED);
```

```
lock(GREEN);
.
.
.
.
.
unlock(GREEN);
```

```
lock(RED);
lock(GREEN);
.
.
.
.
unlock(GREEN);
unlock(RED);
unlock(RED);
```
Non-locality in analysis of nested locks

Worst case cannot be determined by considering each processor independently.
Non-locality in analysis of nested locks

Worst case cannot be determined by considering each processor independently.

Blocking analysis is non-local problem!
Blocking analysis is non-local problem!
Similarities to Matching

Blocking analysis is non-local problem!

Blocking Analysis at least as hard as Multiple Choice Matching
The Multiple-Choice Matching Problem

Input:
  • graph $G=(V,E)$
  • edge partitions $E_1,\ldots,E_t$

Problem:
Is there a matching $F$ that contains exactly one edge from each edge partition?
The Multiple-Choice Matching Problem

Input:

Problem:
Is there a matching $F$ that contains exactly one edge from each edge partition?

Problem:
Is there a matching $F$ that contains exactly one edge from each edge partition?
The Multiple-Choice Matching Problem

Input:

Problem:
Is there a matching $F$ that contains exactly one edge from each edge partition?
The Multiple-Choice Matching Problem

Input:

Problem:
Is there a matching \( F \) that contains exactly one edge from each edge partition?

NP-HARD
Outline

- Introduction ✔
- Intuition: Why does nesting make the analysis difficult? ✔
- Reduction: From Multiple Choice Matching to Blocking Analysis ✔
- Summary and Conclusion
The Multiple-Choice Matching Problem

Input:

Problem:
Is there a matching $F$ that contains exactly one edge from each edge partition?

Problem:
Is there a matching $F$ that contains exactly one edge from each edge partition?
Reduction Overview

Multiple-Choice Matching

Blocking analysis for nested locks

polynomial-time transformations

Oracle
Reduction Overview

Multiple-Choice Matching

polynomial-time transformations

Blocking analysis for nested locks
Reduction Overview

- encode MCM problem as blocking analysis problem

- Multiple-Choice Matching
  - polynomial-time transformations

- Blocking analysis for nested locks
Encoding Vertices as Resources

Multiple-Choice Matching Problem

Blocking Analysis Problem

1 ➞ 1

2 ➞ 2

vertices ➞ resources
Encoding Vertices as Resources

Multiple-Choice Matching Problem

Blocking Analysis Problem

critical section for resource 1

vertices ➞ resources

1

2
Encoding Edges as Nested Critical Sections

Multiple-Choice Matching Problem

Blocking Analysis Problem

edges $\rightarrow$ outer critical sections
Encoding Edges as Nested Critical Sections

Multiple-Choice Matching Problem

Blocking Analysis Problem

critical section for resource D with nested critical sections for 1 and 2

edges → outer critical sections
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Blocking Analysis Problem

CPU 1

D

1

2

CPU 2

D

2

3

D

3

4
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Blocking Analysis Problem

CPU 0

CPU 1

CPU 2

CPU 3

1 2 3 4

D 1 2 3 4

1 2 3 4
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Blocking Analysis Problem

“probe” critical section
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Blocking Analysis Problem

one long and one short critical section for each shared resource
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Blocking Analysis Problem

CPU 0 | CPU 1 | CPU 2 | CPU 3
---|---|---|---
D | 1 | 1 | 1
D | 2 | 2 | 2
D | 3 | 3 | 3
D | 4 | 4 | 4
Is there a set of edges $F$ such that $F$ contains exactly one green and one purple edge.
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

Blocking Analysis Problem

CPU 0

CPU 1

CPU 2

CPU 3
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

and

Is there a worst-case schedule such that

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

Blocking Analysis Problem

Is there a worst-case schedule such that

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

Blocking Analysis Problem

Is there a worst-case schedule such that

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks
Example: MCM Solution \textbf{Does} Exist

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- All edges in $F$ are pairwise disjoint?

- $F$ contains exactly one green and one purple edge?

Blocking Analysis Problem

Is there a worst-case schedule such that

- Exactly one CS for $D$ from CPU 1 and CPU 2 block?

- No short CS from CPU 3 transitively blocks?
Example: MCM Solution **Does Exist**

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- All edges in $F$ are pairwise disjoint
- $F$ contains exactly one green and one purple edge

Blocking Analysis Problem

Is there a worst-case schedule such that

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks

**Example: MCM Solution Does Exist**

- CPU 0
- CPU 2
- CPU 1
- CPU 3

Diagram showing the scheduling and blocking analysis.
Example: MCM Solution **Does Exist**

**Multiple-Choice Matching Problem**

Is there a set of edges $F$ such that:

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

**Blocking Analysis Problem**

Is there a worst-case schedule such that:

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks

---

Example: MCM Solution **Does Exist**
Example: MCM Solution Does Exist

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that $F$ contains exactly one green and one purple edge and all edges in $F$ are pairwise disjoint.

Blocking Analysis Problem

Is there a worst-case schedule such that exactly one CS for $D$ from CPU 1 and CPU 2 block, no short CS from CPU 3 transitively blocks.
Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

Blocking Analysis Problem

Is there a worst-case schedule such that

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks

Example: MCM Solution **Does** Exist
Example: MCM Solution **Does Exist**

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that:

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

Blocking Analysis Problem

Is there a worst-case schedule such that:

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks
Example: MCM Solution **Does Not Exist**
Example: MCM Solution **Does Not Exist**

### Multiple-Choice Matching Problem

Is there a set of edges $F$ such that:

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

### Blocking Analysis Problem

Is there a worst-case schedule such that:

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks
Example: MCM Solution **Does Not Exist**

**Multiple-Choice Matching Problem**

Is there a set of edges $F$ such that:

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

**Blocking Analysis Problem**

Is there a worst-case schedule such that:

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks
Example: MCM Solution Does Not Exist

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that

- $F$ contains exactly one green and one purple edge
- all edges in $F$ are pairwise disjoint

Blocking Analysis Problem

Is there a worst-case schedule such that

- exactly one CS for $D$ from CPU 1 and CPU 2 block
- no short CS from CPU 3 transitively blocks
Example: MCM Solution Does Not Exist

Multiple-Choice Matching Problem

Is there a set of edges $F$ such that $F$ contains exactly one green and one purple edge and all edges in $F$ are pairwise disjoint.

Blocking Analysis Problem

Is there a worst-case schedule such that exactly one CS for $D$ from CPU 1 and CPU 2 block and no short CS from CPU 3 transitively blocks.
Example: MCM Solution **Does Not Exist**

**Multiple-Choice Matching Problem**

Is there a set of edges $F$ such that:

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

**Blocking Analysis Problem**

Is there a worst-case schedule such that:

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks
Example: MCM Solution **Does Not Exist**

**Multiple-Choice Matching Problem**

Is there a set of edges $F$ such that

- $F$ contains exactly one green and one purple edge
- All edges in $F$ are pairwise disjoint

**Blocking Analysis Problem**

Is there a worst-case schedule such that

- Exactly one CS for $D$ from CPU 1 and CPU 2 block
- No short CS from CPU 3 transitively blocks
Encoding MCM as a Blocking Analysis

Result of blocking analysis is bound on blocking duration!
Encoding MCM as a Blocking Analysis

Is there a worst-case schedule such that exactly one CS for D from CPU 1 and CPU 2 block and no short CS from CPU 3 transitively blocks?

Result of blocking analysis is bound on blocking duration!
Encoding MCM as a Blocking Analysis

Result of blocking analysis is bound on blocking duration!

Is there a worst-case schedule such that

- exactly one CS for D from CPU 1 and CPU 2 block
- no short CS from CPU 3 transitively blocks

How to derive from blocking bound whether conditions hold?
Encoding MCM as a Blocking Analysis

Blocking Analysis Problem

CPU 0

D

CPU 1

D

1
2

CPU 2

D

2
3

CPU 3

1
2
3
4

3
4
Encoding MCM as a Blocking Analysis

Blocking Analysis Problem

Choose critical section lengths such that it can be inferred from the blocking bound:

• how many requests from CPU 1 and CPU 2 block, and
• how many short requests from CPU 3 transitively block.
Encoding MCM as a Blocking Analysis

Blocking Analysis

Choose critical section lengths such that it can be inferred from the blocking bound:

- how many requests from CPU 1 and CPU 2 block, and
- how many short requests from CPU 3 transitively block.

See paper for details!
Generality of Reduction

CPU 0

CPU 1

CPU 2

CPU 3

D

1

2

D

2

3

D

3

4

D

1

2

3

4

1

2

3

4
Generality of Reduction

Reduction uses a single job per processor.
Generality of Reduction

Reduction uses a single job per processor.

Reduction oblivious to

- spin based vs. suspension based locks
- preemptable vs. non-preemptable spinning
- any work conserving scheduler
Outline

• Introduction ✓
• Intuition: Why does nesting make the analysis difficult? ✓
• Reduction: From Multiple Choice Matching to Blocking Analysis ✓
• Summary and Conclusion
Summary

- multiprocessor system
- locks with strong ordering guarantees (FIFO/Priority)
- nested critical sections

Blocking analysis
NP-hard!
Summary

Multiprocessor system

Locks with strong ordering guarantees (FIFO/Priority)

Nested critical sections

Blocking analysis

NP-hard!

Reduction from Multiple Choice Matching Problem
Future Work

FIFO/priority-ordered locks: Approximation hardness? PTAS?
Future Work

FIFO/priority-ordered locks: Approximation hardness? PTAS?

Nesting and efficient analysis possible?
This slide is intentionally left blank.
PTAS for Blocking Analysis of Nested Locks

3-SAT

polynomial-time transformation

Multiple-choice matching

polynomial-time transformation

Blocking analysis for nested spin locks
PTAS for Blocking Analysis of Nested Locks

- 3-SAT
  - Polynomial-time transformation
  - Multiple-choice matching
  - Polynomial-time transformation
  - Blocking analysis for nested spin locks
- Max 3-SAT
PTAS for Blocking Analysis of Nested Locks

3-SAT

Max 3-SAT

Multiple-choice matching

Blocking analysis for nested spin locks

APX-complete: PTAS does not exist

3-SAT \rightarrow \text{polynomial-time transformation} \rightarrow \text{Multiple-choice matching} \rightarrow \text{Blocking analysis for nested spin locks} \rightarrow \text{Max 3-SAT}
PTAS for Blocking Analysis of Nested Locks

3-SAT

Multiple-choice matching

Blocking analysis for nested spin locks

Max 3-SAT

APX-complete: PTAS does not exist

polynomial-time transformation