Computing the Maximum Blocking Time for Scheduling with Deferred Preemption

Sebastian Altmeyer, Claire Burguière, Reinhard Wilhelm

Saarland University

STFSSD, Tokyo 2009
Motivation

preemptive schedule

good performance/bad predictability

non-preemptive schedule

bad performance/good predictability

tradeoff:

defered preemption

**deferred preemption:**
preemption only at predefined, selected preemption points
Schedulability Analysis for Deferred Preemption

Schedulability needs:

- Periods
- Worst Case Execution Time
- Context Switch Costs
- Maximum Blocking Time

\[ T_1 \quad BT \quad BT \]

\[ T_2 \quad \downarrow \quad \text{Context Switch Costs} \quad \downarrow \quad \text{Task Activation} \quad * \quad \text{Preemption Point} \]
Overview

Maximum Blocking Time (MBT) = longest non-preemptive path

1 Basics
   WCET Analysis
   Implicit Path Enumeration Technique
   Cache-Related Preemption Delay

2 Computing Maximum Blocking Time

3 Conclusion
WCET Analysis

Value Analysis determines values for registers and memory cells
Loop Analysis determines loop-bounds
Low-Level Analysis determines WCET for basic blocks
Path Analysis determines longest path through the program
Implicit Path Enumeration Technique

\[ 1 = x_1 \quad x_1 = y_1 + y_2; \]
\[ y_1 + y_4 = x_2 \quad x_2 = y_3 + y_5; \]
\[ y_2 = x_3 \quad x_3 = y_6; \]
\[ y_3 = x_4 \quad x_4 = y_4; \]
\[ y_5 + y_6 = x_5 \quad x_5 = 1; \]
\[ y_3 \leq b_l \cdot y_1; \]

\[ \max \sum_i x_i c_i \]

- node counter \( x_i \), edge counter \( y_i \)
- first node \( x_1 \) entered once, last node \( x_5 \) left once
- nodes executed as often as entered/left
- loop \( l \) executed \( b_l \) times as often as entered
- WCET given by max. sum of \( x_i \) times \( c_i \) (\( = \) WCET of node \( i \))
- set of constraints solved using ILP
Cache-Related Preemption Delay (CRPD)

At basic block (2):
- \(a\) may be cached, may be reused at (5)
- \(d\) may be cached, may be reused at (4)

Preemption at (2) \(\Rightarrow\) 2 additional misses may occur
Useful Cache Blocks

A memory block is a useful cache block (UCB) at program point \( P \), if it

- may be cached at \( P \), and
- may be reused at program point \( P' \) reached from \( P \).

CRPD at basic block \( i \):

\[
CRPD_i \leq cost_{\text{reload}} \times |UCB_i|.
\]
Maximum Blocking Time

Maximum Blocking Time (MBT) = Longest Path from preemption point or program start to preemption point or program end

longest path given by:
WCET of the basic blocks + CRPD

for the sake of simplicity:
preemption points only at the beginning or end of a basic block
Maximum Blocking Time

1. partition preemption points into two sets:
   \[ Up = \{ i \mid \text{preemption point at beginning of } B_i \} \]
   \[ Lo = \{ i \mid \text{preemption point at end of } B_i \} \]

2. introduce artificial nodes:

   \[ Up = \{ 3 \} \quad Lo = \{ 4 \} \]
• theoretical start node $S$, connected to beginnings of all paths
• theoretical end node $E$, connected to ends of all paths
• structural constraints as before (but with additional nodes)
• MBT given by max. $\sum x_i c_i + \text{possible CRPD } C_i$
• theoretical start node $S$, connected to beginnings of all paths
• theoretical end node $E$, connected to ends of all paths
• structural constraints as before (but with additional nodes)
• MBT given by max. $\sum x_i c_i$ + possible CRPD $C_i$
\[ x_1 + x_3 + x_4^l = 1; \]
\[ 1 = x_4 + x_5 + x_3^u; \]

- theoretical start node \( S \), connected to beginnings of all paths
- theoretical end node \( E \), connected to ends of all paths
- structural constraints as before (but with additional nodes)
- MBT given by max. \( \sum x_i c_i \) + possible CRPD \( C_i \)
The diagram shows a network with nodes labeled from 1 to 5, connected by edges labeled with variables. The system is described by the following equations:

- $x_1 + x_3 + x_4^l = 1$
- $1 = x_4 + x_5 + x_4^u$
- $x_1 = y_1 + y_2$
- $y_1 + y_4 = x_2$
- $x_2 = y_3 + y_5$
- $y_2 = x_3^u$
- $x_3 = y_6$
- $y_3 = x_4$
- $x_4^l = y_4$
- $y_5 + y_6 = x_5$
- $y_3 \leq b y_1$

The network includes:

- **Theoretical start node** $S$, connected to the beginnings of all paths
- **Theoretical end node** $E$, connected to the ends of all paths
- **Structural constraints** as before (but with additional nodes)
- **MBT** given by max. $\sum x_i c_i +$ possible CRPD $C_i$
\[
\begin{align*}
x_1 + x_3 + x_4^l &= 1; \\
1 &= x_4 + x_5 + x_3^u; \\
x_1 &= y_1 + y_2; \\
y_1 + y_4 &= x_2 & x_2 &= y_3 + y_5; \\
y_2 &= x_3^u; & x_3 &= y_6; \\
y_3 &= x_4; & x_4^l &= y_4; \\
y_5 + y_6 &= x_5; \\
y_3 &\le b y_1; \\
\end{align*}
\]

- theoretical start node $S$, connected to beginnings of all paths
- theoretical end node $E$, connected to ends of all paths
- structural constraints as before (but with additional nodes)
- MBT given by max. $\sum x_i c_i + x_4^l C_4^l + x_3 C_3^u$
Conclusion

• Essential ingredient for the analysis of deferred preemption
• Only minor extensions to IPET:
  • one new variable per preemption point
  • no additional constraints
• All data available from WCET / UCB analysis

& Future Work

• implementation and evaluation
• automatic derivation of good preemption points
Implicit Path Enumeration - Equations

\[
\text{max} \sum_{i} x_i c_i \quad (1)
\]

\[x_i = 1, \text{ where } B_i = s \quad (2)\]

\[x_j = 1, \text{ where } B_j = e \quad (3)\]

\[\sum_{j \in S} y_j = x_k, \text{ where } S = \{j|E_j = (\_ , B_k)\} \quad (4)\]

\[x_k = \sum_{j \in S} y_j, \text{ where } S = \{j|E_j = (B_k , \_ )\} \quad (5)\]

\[y_l \leq b_l (\sum_{j \in S} y_j) \text{ where } S = \{j|E_j \text{ loop entry edge}\} \quad (6)\]
Maximum Blocking Time - Equations 1

\[
\max \sum_{i=1}^{n} (x_i c_i) + \sum_{B_i \in Lo} x_i^l C_i^l + \sum_{B_i \in Up} x_i C_i^u
\]

\[
x_s + \sum_{B_i \in Lo} x_i^l + \sum_{B_i \in Up} x_i = 1
\]

\[
x_e + \sum_{B_i \in Up} x_i^u + \sum_{B_i \in Lo} x_i = 1
\]
Maximum Blocking Time - Equations 2

\[
\sum_{j \in S} y_j = \begin{cases} 
  x_k^u & \text{if } B_k \in Up \\
  x_k & \text{otherwise}
\end{cases} 
\]  
(10)

where \( S = \{j | E_j = (\_, B_k)\} \)

\[
\sum_{j \in S} y_j = \begin{cases} 
  x_k^l & \text{if } B_k \in Lo \\
  x_k & \text{otherwise}
\end{cases} 
\]  
(11)

where \( S = \{j | E_j = (B_k, \_)\} \)

\[
y_l \leq b_l \left( \sum_{j \in S} y_j \right) 
\]  
(12)

where \( S = \{j | E_j \text{ loop entry edge}\} \)