Cache Related Preemption Delay for Set-Associative Caches
Resilience Analysis

Sebastian Altmeyer, Claire Burguière, Jan Reineke

AVACS Workshop, Oldenburg 2009
Why use preemptive scheduling?

- Preemption often increases schedulability of task sets.
- Tasks with short deadlines are often unschedulable non-preemptively.

Example

Given: Two periodic tasks $T_1$ and $T_2$, with periods $P_1 = 2$, $P_2 = 8$, deadlines $D_1 = P_1$, $D_2 = P_2$, and execution times $C_1 = 1$, $C_2 = 3$. 
Why use preemptive scheduling?

- Preemption often increases schedulability of task sets.
- Tasks with short deadlines are often unschedulable non-preemptively.

Example

Given: Two periodic tasks $T_1$ and $T_2$, with periods $P_1 = 2$, $P_2 = 8$, deadlines $D_1 = P_1$, $D_2 = P_2$, and execution times $C_1 = 1$, $C_2 = 3$. 
Preemption does not come for free!

- The preemption task "disturbs" the state of performance-enhancing features like caches and pipelines.
- Once the preempted task resumes its execution, the disturbance may cause additional cache misses.
- The additional execution time due to additional cache misses is known as the cache-related preemption delay (CRPD).

\[ T_2 \uparrow \quad T_1 \quad \text{CRPD} \]

\[ \uparrow \quad \text{Task Activation} \]
How to take preemption cost into account?

Where to account for preemption cost?

- Integrate into WCET Analysis: [?]  
  - assume cache misses everywhere  
  - very pessimistic but easy to use in schedulability analysis

- WCET Analysis + CRPD Analysis: [?]  
  - $WCET_{bound} + n \cdot CRPD_{bound} \geq$ execution time of task with up to $n$ preemptions  
  - more precise but not supported by many schedulability analyses
CRPD for set-associative caches - LRU

■ CRPD computation:
  ▶ Preempted task: Useful Cache Blocks (UCB)
  ▶ Preempting task: Evicting Cache Blocks (ECB)

■ CRPD from UCB and ECB:
  ▶ Previous combination rather imprecise
  ⇒ Some UCBs remain useful under preemption
Useful Cache Block - [?]  

Definition (Useful Cache Block)

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

a) \( m \) may be cached at \( P \)

b) \( m \) may be reused at program point \( P' \) that may be reached from \( P \) with no eviction of \( m \) on this path.
**Useful Cache Block - [?]**

**Definition (Useful Cache Block)**

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

a) \( m \) may be cached at \( P \)

b) \( m \) may be reused at program point \( P' \) that may be reached from \( P \) with no eviction of \( m \) on this path.

\( \times = \text{hit} \)
\( \bigcirc = \text{miss} \)

Cache Content: \([A, B, C, D]\)
Useful Cache Block - [?]  

Definition (Useful Cache Block)

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

a) \( m \) may be cached at \( P \)

b) \( m \) may be reused at program point \( P' \) that may be reached from \( P \) with no eviction of \( m \) on this path.

\[ \begin{align*}
\times & = \text{hit} \\
\circ & = \text{miss}
\end{align*} \]

\[ \text{Cache Content: } [A, B, C, D] \]

\[ \begin{align*}
\text{CRPD}_{\text{UCB}} & = \sum_{s=1}^{c} \text{CRPD}_{\text{UCB}}^s \\
\text{CRPD}_{\text{UCB}}^s & = \text{BRT} \times \min(\| \text{UCB}(s) \|, n)
\end{align*} \]

\( n = \text{associativity} \)

\( \text{BRT} = \text{Block Reload Time} \)
Evicting Cache Blocks

Definition (Evicting Cache Blocks (ECB))

A memory block of the preemtping task is called an evicting cache block, if it may be accessed during the execution of the preemtping task.

Cache Content: \([A, B, C, D]\)

Cache Content: \([X, Y, Z, D]\)

\(\circ\) = additional miss due to preemption (CRPD)
Evicting Cache Blocks

Definition (Evicting Cache Blocks (ECB))

A memory block of the preempting task is called an evicting cache block, if it may be accessed during the execution of the preempting task.

$\text{CRPD}_{\text{ECB}}^s = \begin{cases} 0 & \text{if } \text{ECB}(s) = \emptyset \\ \text{BRT} \times n & \text{otherwise} \end{cases}$
Impact of the preemtting task on the preempted task

CRPD (using UCB and ECB)

\[ CRPD_{UCB\&ECB} = \sum_{s=1}^{c} \min(CRPD_{UCB}^s, CRPD_{ECB}^s) \]
Impact of the preemtting task on the preempted task: Example

\[
\begin{align*}
[c, b, a, x] & \xrightarrow{a} [a, c, b, x] & b & [b, a, c, x] & c & [c, b, a, x] & \text{no misses}
\end{align*}
\]
Impact of the preemtting task on the preempted task: Example

\[
\begin{align*}
\text{ECBs} &= \{e\} \\
[c, b, a, x] &\xrightarrow{a} [a, c, b, x] \xrightarrow{b} [b, a, c, x] \xrightarrow{c} [c, b, a, x] \quad \text{no misses} \\
[e, c, b, a] &\xrightarrow{a} [a, e, c, b] \xrightarrow{b} [b, a, e, c] \xrightarrow{c} [c, b, a, e] \quad \text{no misses}
\end{align*}
\]

- \( \text{CRPD}_{\text{UCB}} \Rightarrow |\text{UCB}| = 3 \)
- \( \text{CRPD}_{\text{ECB}} \Rightarrow n = 4 \)
- \( \text{CRPD}_{\text{UCB}\&\text{ECB}} = \min(\text{CRPD}_{\text{UCB}}, \text{CRPD}_{\text{ECB}}) \Rightarrow 3 \)
  - Overestimation: number of additional misses = 0 < 3
Impact of the preempting task on the preempted task: Example

ECBs = \{e\}

\[ [c, b, a, x] \xrightarrow{a} [a, c, b, x] \xrightarrow{b} [b, a, c, x] \xrightarrow{c} [c, b, a, x] \text{ no misses} \]

\[ [e, c, b, a] \xrightarrow{a} [a, e, c, b] \xrightarrow{b} [b, a, e, c] \xrightarrow{c} [c, b, a, e] \text{ no misses} \]

- \( \text{CRPD}_{\text{UCB}} \Rightarrow |\text{UCB}| = 3 \)
- \( \text{CRPD}_{\text{ECB}} \Rightarrow n = 4 \)
- \( \text{CRPD}_{\text{UCB}} \& \text{ECB} = \min(\text{CRPD}_{\text{UCB}}, \text{CRPD}_{\text{ECB}}) \Rightarrow 3 \)
  - Overestimation: number of additional misses = 0 < 3

- Why?
  - \(|\text{ECB}| = 2\), but
  - \(|\text{ECB}| = 1\)
  - A single ECB is not sufficient to evict a UCB.
Combining UCB and ECB: Refinement

Determining \( \max |ECB| \), such that no additional cache miss occur
Combining UCB and ECB: Refinement

Determining $\max |\text{ECB}|$, such that no additional cache miss occur

$m \in \text{UCB}$

$m$ is 4-resilient
Definition (l-Resilience)

A memory block \( m \) is called \( l \)-resilient at program point \( P \), if all possible next accesses to \( m \)

- that would be hits without preemption,
- would still be hits in case of a preemption at \( P \) with \( l \) accesses.
Resilience analysis

Definition (l-Resilience)

A memory block \( m \) is called \( l \)-resilient at program point \( P \), if all possible next accesses to \( m \)

- *that would be hits without preemption*,
- *would still be hits in case of a preemption at \( P \) with \( l \) accesses.*

- No UCB is \( n \)-resilient, i.e., no UCB remains useful after a preemption with \( n \) ECBs.
- Each \( (l + 1) \)-resilient UCB is also \( l \)-resilient.
- Each UCB is at least \( 0 \)-resilient.
Resilience analysis

Definition (l-Resilience)

A memory block \( m \) is called \( l \)-resilient at program point \( P \), if all possible next accesses to \( m \) that would be hits without preemption, would still be hits in case of a preemption at \( P \) with \( l \) accesses.

\[ m \in UCB \]

\( m \) is 4-resilient

\[ ECB = \{e_1, e_2, e_3, e_4\} \]
Resilience analysis

Definition (l-Resilience)

A memory block \( m \) is called \( l \)-resilient at program point \( P \), if all possible next accesses to \( m \)

- that would be hits without preemption,
- would still be hits in case of a preemption at \( P \) with \( l \) accesses.

\[ m \in UCB \]
\[ m \text{ is } 4\text{-resilient} \]

\[ ECB = \{ e_1, e_2, e_3, e_4 \} \]

In general: if \( |ECB| \leq l \) then the UCB is not evicted
Resilience analysis

Definition (l-Resilience)

A memory block $m$ is called $l$-resilient at program point $P$, if all possible next accesses to $m$
- that would be hits without preemption,
- would still be hits in case of a preemption at $P$ with $l$ accesses.

0-resil.  2-resil.  m
0-resilient.  2-resil.  m

no access to $m$  m is not useful

no access to $m$  m

m  m  m
Bounding the CRPD using Resilience

CRPD (Combining UCB and ECB by using Resilience)

\[
\text{blocks contributing to CRPD} = \underbrace{\text{UCB} \setminus \{m \mid m \text{ is ECB-resilient}\}}_{\text{useful}} \cup \underbrace{\{m \mid m \text{ is ECB-resilient}\}}_{\text{remain useful}}
\]
Bounding the CRPD using Resilience

CRPD (Combining UCB and ECB by using Resilience)

\[ CRPD \leq BRT \times \left| \frac{UCB}{useful} \setminus \{ m \mid m \text{ is } ECB{-}\text{-resilient} \} \right| \]

blocks contributing to CRPD

remain useful
Bounding the CRPD using Resilience: Example

ECBs = \{ e \}

\[
\begin{align*}
[c, b, a, x] & \xrightarrow{a} [a, c, b, x] \xrightarrow{b} [b, a, c, x] \xrightarrow{c} [c, b, a, x] & \text{no misses} \\
[e, c, b, a] & \xrightarrow{a} [a, e, c, b] \xrightarrow{b} [b, a, e, c] \xrightarrow{c} [c, b, a, e] & \text{no misses}
\end{align*}
\]
Bounding the CRPD using Resilience: Example

- $|\text{ECB}| = 1$
- $a$, $b$ and $c$ are $1$-resilient
- $CRPD_{\text{UCB}\&\text{ECB}}^{\text{res}} = BRT \times |UCB \setminus \{m \mid m \text{ is } |\text{ECB}|\text{-resilient}\}| = 0$
Bounding the CRPD using Resilience: Example

- ECBs
  - \( \{ e \} \)

- No misses

\[
\begin{align*}
[c, b, a, x] & \xrightarrow{a} [a, c, b, x] & \xrightarrow{b} [b, a, c, x] & \xrightarrow{c} [c, b, a, x] & \text{no misses} \\
[e, c, b, a] & \xrightarrow{a} [a, e, c, b] & \xrightarrow{b} [b, a, e, c] & \xrightarrow{c} [c, b, a, e] & \text{no misses}
\end{align*}
\]

- |ECB| = 1
- \( a, b \) and \( c \) are 1-resilient
- \( CRPD_{U CB \& E CB}^{res} = BRT \times |U CB \setminus \{ m \mid m \text{ is } |ECB|-\text{resilient} \}| = 0 \)

- Instead of: \( CRPD_{U CB \& E CB} = min(CRPD_{U CB}, CRPD_{E CB}) = 3 \times BRT \)
Bounding the CRPD using Resilience: Example

- $|\text{ECB}| = 1$
- $a, b$ and $c$ are 1-resilient
- $CRPD_{\text{UCB,ECD}}^{\text{res}} = BRT \times |\text{UCB} \setminus \{m \mid m \text{ is } |\text{ECB}|\text{-resilient}\}| = 0$

Instead of: $CRPD_{\text{UCB,ECD}} = \min(CRPD_{\text{UCB}}, CRPD_{\text{ECD}}) = 3 \times BRT$
Conclusions

- Preemptive scheduling:
  - sometimes necessary
  - but not for free: CRPD

- UCB and ECB analyses:
  - pessimistic overapproximation of the CRPD

- Resilience analysis:
  - determining the set of UCBs that remain useful under preemption
  - increase precision
  - implemented as two simple data-flow analyses:
    - similar to UCB analysis for LRU
    - currently in the phase of evaluation
Further reading

A New Notion of Useful Cache Block to Improve the Bounds of Cache-Related Preemption Delay.

Analysis of cache-related preemption delay in fixed-priority preemptive scheduling.
In RTSS’96 p. 264, IEEE Computer Society.

Accurate estimation of cache-related preemption delay.
In CODES+ISSS’03 ACM.

Caches in WCET Analysis.

Schneider, J. (2000).
Cache and pipeline sensitive fixed priority scheduling for preemptive real-time systems.

Scalable precision cache analysis for real-time software.
ACM TECS 6, 25.

Integrated intra- and inter-task cache analysis for preemptive multi-tasking real-time
/-resilience analysis

(a) 0-resilient

(b) m is not useful 2-resilient

2-resilient
CPRD using ECB: Pitfall

ECBs = \{e\}

\[ [b, a, 9, 8] \xrightarrow{8} [8, b, a, 9] \xrightarrow{9} [9, 8, b, a] \xrightarrow{a} [a, 9, 8, b] \xrightarrow{b} [b, a, 9, 8] \]

0 misses

\[ [e, b, a, 9] \xrightarrow{8^*} [8, e, b, a] \xrightarrow{9^*} [9, 8, e, b] \xrightarrow{a^*} [a, 9, 8, e] \xrightarrow{b^*} [b, a, 9, 8] \]

4 misses

- \(|UCB(s)| = 4\)
- \(|ECB(s)| = 1\)
- \(n = 4\)
- number of additional misses = 4
Upper-bound on the CRPD - direct-mapped caches

- using UCB [?]:
  \[
  \text{CRPD}_{\text{UCB}} = \text{BRT} \cdot |\{ s_i \mid \exists m \in \text{UCB} : m \mod c = s_i \}|
  \]

- using ECB [?]:
  \[
  \text{CRPD}_{\text{ECB}} = \text{BRT} \cdot |\{ s_i \mid \exists m \in \text{ECB} : m \mod c = s_i \}|
  \]

- using UCB and ECB [? , ?]:
  \[
  \text{CRPD}_{\text{UCB}\&\text{ECB}} = \text{BRT} \cdot |\{ s_i \mid \exists m \in \text{UCB} : m \mod c = s_i \\
  \quad \land \exists m' \in \text{ECB} : m' \mod c = s_i \}|
  \]
CRPD for FIFO: Pitfalls

ECBs

\[
\begin{align*}
[b, a] & \xrightarrow{a} [b, a] & \xrightarrow{e^*} [e, b] & \xrightarrow{b} [e, b] & \xrightarrow{c^*} [c, e] & \xrightarrow{e} [c, e] & \text{2 misses} \\
[x, b] & \xrightarrow{a^*} [a, x] & \xrightarrow{e^*} [e, a] & \xrightarrow{b^*} [b, e] & \xrightarrow{c^*} [c, b] & \xrightarrow{e^*} [e, c] & \text{5 misses}
\end{align*}
\]
CRPD for FIFO: Pitfalls

ECBs
= \{x\}

\[
\begin{align*}
[b, a] & \xrightarrow{a} [b, a] \xrightarrow{e^*} [e, b] \xrightarrow{b} [e, b] \xrightarrow{c^*} [c, e] \xrightarrow{e} [c, e] & 2 \text{ misses} \\
[x, b] & \xrightarrow{a^*} [a, x] \xrightarrow{e^*} [e, a] \xrightarrow{b^*} [b, e] \xrightarrow{c^*} [c, b] \xrightarrow{e^*} [e, c] & 5 \text{ misses}
\end{align*}
\]

\begin{itemize}
  \item |UCB(s)| = 2
  \item |ECB(s)| = 1
  \item n = 2
  \item But: number of additional misses = 3
\end{itemize}
CRPD for PLRU: Pitfalls

- |UCB(s)| = 4
- |ECB(s)| = 2
- n = 4
- But: number of additional misses = 5