Cache Related Preemption Delay for Set-Associative Caches
Pitfalls and Solutions

Claire Burguière, Jan Reineke, Sebastian Altmeyer

Workshop on WCET Analysis, Dublin 2009
Preemptive scheduling

Cache related preemption delay (CRPD):
  ▶ Impact of preemption on the cache content
  ▶ Overall cost of additional reloads due to preemption

\[ T_1 = \text{CRPD} = T_2 \]

\[ = \text{CRPD} \]

\[ = \text{Task Activation} \]
CRPD computation for set-associative caches

- CRPD computation:
  - Useful Cache Blocks (UCB)
  - Evicting Cache Blocks (ECB)

- CRPD for set-associative caches:
  \[\Rightarrow\text{Pitfalls:}\]
  - LRU: CRPD not bounded by the number of ECBs
  - FIFO and PLRU: CRPD not bounded
Useful Cache Block - [Lee et al., 1996]

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.

$x =$ hit
○ = miss

Cache Content: $[A, B, C, D]$

CRPD for set-associative caches

Burguière, Reineke, Altmeyer

WCET, Dublin 2009
Useful Cache Block - [Lee et al., 1996]

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.

$\times =$ hit  
$\bigcirc =$ miss

Cache Content: $[A, B, C, D]$
Useful Cache Block - [Lee et al., 1996]

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.

$\times =$ hit
$\bigcirc =$ miss

CRPD$^{LRU}_{\text{LRU}}(s) = \sum_{s=1}^{c} \text{CRPD}^{LRU}(s)$

CRPD$^{LRU}_{\text{UCB}}(s) = \text{BRT} \cdot \min(|\text{UCB}(s)|, n)$

$n =$ associativity

BRT = Block Reload Time
Evicting Cache Blocks
[Tomiyama & Dutt, 2000]

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.

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

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

\(\bullet = \text{additional miss due to preemption (CRPD)}\)
Evicting Cache Blocks
[Tomiyama & Dutt, 2000]

Definition (Evicting Cache Blocks (ECB))

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

\[
\text{CRPD}_{\text{ECB}}^{\text{LRU}} (s) \equiv \text{BRT} \cdot \min(|\text{ECB}(s)|, n)
\]
CRPD computation for LRU using ECB: Pitfall

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

0 misses
CRPD computation for LRU using ECB: Pitfall

\[ \text{ECBs} = \{ e \} \]

\[
\begin{align*}
[b, a, 9, 8] & \xrightarrow{8} [8, b, a, 9] \xrightarrow{9} [9, 8, b, a] \xrightarrow{a} [a, 9, 8, b] \xrightarrow{b} [b, a, 9, 8] & 0 \text{ misses} \\
[e, b, a, 9] & \xrightarrow{8^*} [8, e, b, a] \xrightarrow{9^*} [9, 8, e, b] \xrightarrow{a^*} [a, 9, 8, e] \xrightarrow{b^*} [b, a, 9, 8] & 4 \text{ misses}
\end{align*}
\]

- \(|\text{UCB}(s)| = 4|
- \(|\text{ECB}(s)| = 1|
- \(n = 4|
- \text{number of additional misses} = 4
ECB derivation used only to know if the set is used by the preempting task:

\[
\text{CRPD}^{\text{LRU}}_{\text{ECB}}(s) = \begin{cases} 
0 & \text{if } \text{ECB}(s) = \emptyset \\
\text{BRT} \cdot n & \text{otherwise}
\end{cases}
\]
CRPD for FIFO: Pitfalls

\[ [b, a] \xrightarrow{a} [b, a] \xrightarrow{e^*} [e, b] \xrightarrow{b} [e, b] \xrightarrow{c^*} [c, e] \xrightarrow{e} [c, e] \text{ 2 misses} \]
CRPD for FIFO: Pitfalls

\[
\begin{align*}
\text{ECBs} &= \{x\} \\
&= \{x\} \\
&\xrightarrow{a} \{x\} \\
&\xrightarrow{a^*} \{x, a\} \\
&\xrightarrow{e^*} \{e, a\} \\
&\xrightarrow{e^*} \{e, b\} \\
&\xrightarrow{b} \{e, b\} \\
&\xrightarrow{c^*} \{c, e\} \\
&\xrightarrow{e} \{c, e\} \\
\end{align*}
\]

2 misses

\[
\begin{align*}
&\xrightarrow{[b, a]} \{b, a\} \\
&\xrightarrow{e^*} \{e, b\} \\
&\xrightarrow{b} \{e, b\} \\
&\xrightarrow{c^*} \{c, e\} \\
&\xrightarrow{e} \{c, e\} \\
\end{align*}
\]

5 misses

- \(|\text{UCB}(s)| = 2|
- \(|\text{ECB}(s)| = 1|
- \(n = 2\)
- But: number of additional misses = 3
Definition – Relative Miss-Competitiveness

Notation

\[ m_P(p, s) = \text{number of misses that policy } P \text{ incurs on access sequence } s \in M^* \text{ starting in state } p \]

Definition (Relative miss competitiveness)

Policy \( P \) is \((k, c)\)-miss-competitive relative to policy \( Q \), if

\[ m_P(p, s) \leq k \cdot m_Q(q, s) + c \]

for all access sequences \( s \in M^* \) and compatible cache-set states \( p, q \).
Definition – Relative Miss-Competitiveness

Notation

\[ m_P(p, s) = \text{number of misses that policy } P \text{ incurs on access sequence } s \in M^* \text{ starting in state } p \]

Definition (Relative miss competitiveness)

Policy \( P \) is \((k, c)\)-miss-competitive relative to policy \( Q \), if

\[ m_P(p, s) \leq k \cdot m_Q(q, s) + c \]

for all access sequences \( s \in M^* \) and compatible cache-set states \( p, q \).

- \( \text{PLRU}(n) \) is \((1, 0)\)-miss-competitive relative to \( \text{LRU}(1 + \log_2 n) \).
- \( \text{FIFO}(n) \) is \( \left(\frac{n}{n-r+1}, r\right)\)-miss-competitive relative to \( \text{LRU}(r) \).
A sequence of memory accesses

Notation:

- \( m \) = number of misses
- \( \bar{m} \) = number of misses in the case of preemption

\[ m_{\text{pre}} = 4 \]
\[ m_{\text{post}} = 2 \]

\[ \bar{m}_{\text{pre}} = m_{\text{pre}} = 4 \]
\[ \bar{m}_{\text{post}} = m_{\text{post}} + m_{\text{CRPD}} = 5 \]
A sequence of memory accesses

- Notation:
  - \( m \) = number of misses
  - \( \overline{m} \) = number of misses in the case of preemption

\[ m_{\text{pre}} = 4 \]
\[ m_{\text{post}} = 2 \]

\[ \overline{m}_{\text{pre}} = m_{\text{pre}} = 4 \]
\[ \overline{m}_{\text{post}} = m_{\text{post}} + m_{\text{CRPD}} = 5 \]

- Relative miss competitiveness:

\[ \overline{m}^P(t) = \overline{m}^P_{\text{pre}} + \overline{m}^P_{\text{post}} \]
A sequence of memory accesses

- Notation:
  - \( m \) = number of misses
  - \( m' \) = number of misses in the case of preemption

\[
\begin{align*}
  m_{\text{pre}} &= 4 \\
  m_{\text{post}} &= 2 \\
\end{align*}
\]

- \( m_{\text{pre}} = m_{\text{pre}} = 4 \)
- \( m_{\text{post}} = m_{\text{post}} + m_{\text{CRPD}} = 5 \)

- Relative miss competitiveness:

\[
\overline{m}^P(t) = \overline{m}^P(t) + \overline{m}^P(t) \\
\leq [k \cdot m_{\text{pre}} + c] + [k \cdot (m_{\text{post}} + m_{\text{CRPD}}) + c]
\]
A sequence of memory accesses

- **Notation:**
  - $m = \text{number of misses}$
  - $\overline{m} = \text{number of misses in the case of preemption}$

$$
\begin{align*}
  \overline{m} & = m_{\text{pre}} = 4 \\
  m_{\text{post}} & = 2 \\
  \overline{m}_{\text{pre}} & = \overline{m}_{\text{pre}} = 4 \\
  \overline{m}_{\text{post}} & = m_{\text{post}} + m_{\text{CRPD}} = 5
\end{align*}
$$

- **Relative miss competitiveness:**

$$
\begin{align*}
  \overline{m}^{P(t)} &= m_{\text{pre}}^{P(t)} + m_{\text{post}}^{P(t)} \\
  &\leq [k \cdot m_{\text{LRU}(s)}^{\text{pre}} + c] + [k \cdot (m_{\text{LRU}(s)}^{\text{post}} + m_{\text{LRU}(s)}^{\text{CRPD}}) + c] \\
  &= [k \cdot (m_{\text{LRU}(s)}^{\text{pre}} + m_{\text{LRU}(s)}^{\text{post}}) + c] + [k \cdot m_{\text{CRPD}}^{\text{LRU}(s)} + c]
\end{align*}
$$
A sequence of memory accesses

- Notation:
  - \( m \) = number of misses
  - \( \overline{m} \) = number of misses in the case of preemption

\[
\begin{align*}
\overline{m}_{pre} &= 4 \\
m_{post} &= 2
\end{align*}
\]

- Relative miss competitiveness:

\[
\overline{m}^P(t) = \overline{m}^P(t)_{pre} + \overline{m}^P(t)_{post}
\]

\[
\leq \ \left[ k \cdot m_{pre}^{LRU(s)} + c \right] + \left[ k \cdot (m_{post}^{LRU(s)} + m_{CRPD}^{LRU(s)}) + c \right]
\]

\[
= \left[ k \cdot (m_{pre}^{LRU(s)} + m_{post}^{LRU(s)}) + c \right] + \left[ k \cdot m_{CRPD}^{LRU(s)} + c \right]
\]

\[
= \left[ k \cdot m^{LRU(s)} + c \right] + \left[ k \cdot m_{CRPD}^{LRU(s)} + c \right]
\]
Relative Competitiveness – Application

- PLRU(8) using LRU(4):
  \[ m^{\text{PLRU}(8)} \leq m^{\text{LRU}(4)} + m^{\text{LRU}(4)} \]

- FIFO(8) using LRU(5):
  \[ m^{\text{FIFO}(8)} \leq (2 \cdot m^{\text{LRU}(5)} + 5) + (2 \cdot m^{\text{LRU}(5)} + 5) \]
Conclusions

<table>
<thead>
<tr>
<th>Pitfalls</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LRU:</strong></td>
<td><strong>LRU:</strong></td>
</tr>
<tr>
<td>$</td>
<td>ECBs</td>
</tr>
<tr>
<td><strong>FIFO and PLRU:</strong></td>
<td>$\Rightarrow \text{ the set is not used}$</td>
</tr>
<tr>
<td>$</td>
<td>UCBs</td>
</tr>
<tr>
<td>\text{ do not bound the number of additional misses}</td>
<td>\text{ using relative competitiveness and LRU bounds}</td>
</tr>
</tbody>
</table>
Further reading

In ECRTS’09.

In RTSS’96 p. 264, IEEE Computer Society.

In CODES+ISSS’03 ACM.

Caches in WCET Analysis.

ACM Trans. on Embedded Computing Sys. 6, 25.

In SCOPES’04 pp. 182–199.

In CODES’00 ACM.
Upper-bound on the CRPD - direct-mapped caches

- using UCB [Lee et al., 1996]:

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

- using ECB [Tomiyama & Dutt, 2000]:

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

- using UCB and ECB [Negi et al., 2003, Tan & Mooney, 2004]:

\[
\text{CRPD}_{\text{UCB}\&\text{ECB}} = BRT \cdot |\{s_i \mid \exists m \in \text{UCB} : m \mod c = s_i \land \exists m' \in \text{ECB} : m' \mod c = s_i\}|
\]
Relative Competitiveness – Example

\[
\overline{m}_{pre}^{P(t)} \leq k \cdot \overline{m}_{pre}^{LRU(s)} + c = k \cdot m_{pre}^{LRU(s)} + c
\]

\[
\overline{m}^{P(t)} = \overline{m}_{pre}^{P(t)} + \overline{m}_{post}^{P(t)} \leq k \cdot m_{pre}^{LRU(s)} + c + k \cdot (m_{post}^{LRU(s)} + m_{CRPD}^{LRU(s)}) + c = (k \cdot (m_{pre}^{LRU(s)} + m_{post}^{LRU(s)}) + c) + (k \cdot m_{CRPD}^{LRU(s)} + c) = (k \cdot m^{LRU(s)} + c) + (k \cdot m_{CRPD}^{LRU(s)} + c).
\]
Relative Competitiveness – Application

- PLRU(n) using LRU(1 + log₂n):
  - \( k = 1, \ c = 0 \)
  \[
  m^{PLRU(n)} \leq m^{LRU(1 + \log_2 n)} + m^{LRU(1 + \log_2 n)}
  \]
  - Example (n=8):
    \[
    m^{PLRU(8)} \leq m^{LRU(4)} + m^{LRU(4)}
    \]

- FIFO(n) using LRU(r):
  - \( k = \frac{n}{n-r+1}, \ c = r \)
  \[
  m^{FIFO(n)} \leq \left( \frac{n}{n-r+1} \cdot m^{LRU(r)} + r \right) + \left( \frac{n}{n-r+1} \cdot m^{LRU(r)} + r \right)
  \]
  - Example (n=8, r=5):
    \[
    m^{FIFO(8)} \leq \left( 2 \cdot m^{LRU(5)} + 5 \right) + \left( 2 \cdot m^{LRU(5)} + 5 \right)
    \]
CRPD for PLRU: Pitfalls

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