

# Cache Related Preemption Delay for Set-Associative Caches

Resilience Analysis

Sebastian Altmeyer, Claire Burguière, Jan Reineke

AVACS Workshop, Oldenburg 2009

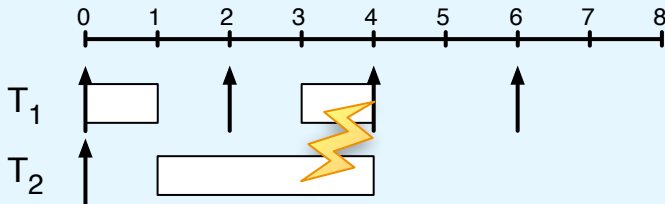


# Why use preemptive scheduling?

- 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  $C_1 = 1$ ,  $C_2 = 3$ .

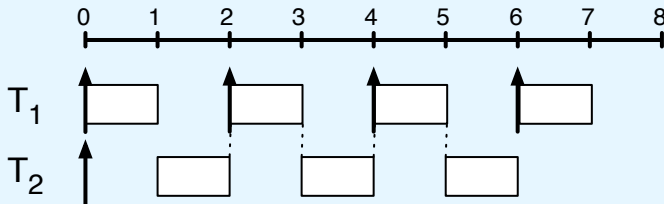


# Why use preemptive scheduling?

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

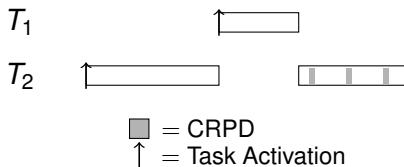
## Example

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 preempting 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).



## 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 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

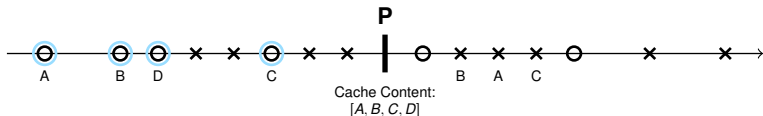
# Useful Cache Block - [?]

## Definition (Useful Cache Block)

A memory block  $m$  at program point  $P$  is called a useful cache block, if

- $m$  may be cached at  $P$
- $m$  may be reused at program point  $P'$  that may be reached from  $P$  with no eviction of  $m$  on this path.

x = hit  
O = miss



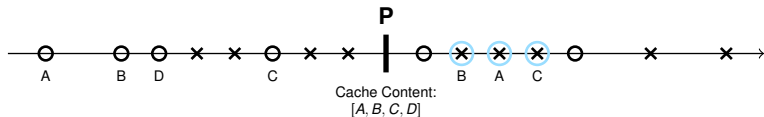
# Useful Cache Block - [?]

## Definition (Useful Cache Block)

A memory block  $m$  at program point  $P$  is called a useful cache block, if

- $m$  may be cached at  $P$
- $m$  may be reused at program point  $P'$  that may be reached from  $P$  with no eviction of  $m$  on this path.

x = hit  
O = miss





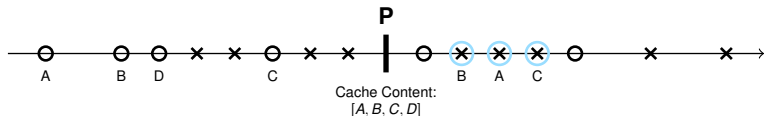
# Useful Cache Block - [?]

## Definition (Useful Cache Block)

A memory block  $m$  at program point  $P$  is called a useful cache block, if

- $m$  may be cached at  $P$
- $m$  may be reused at program point  $P'$  that may be reached from  $P$  with no eviction of  $m$  on this path.

x = hit  
O = miss



$$\text{CRPD}_{\text{UCB}} = \sum_{s=1}^c \text{CRPD}_{\text{UCB}}^s$$

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

$n$  = associativity

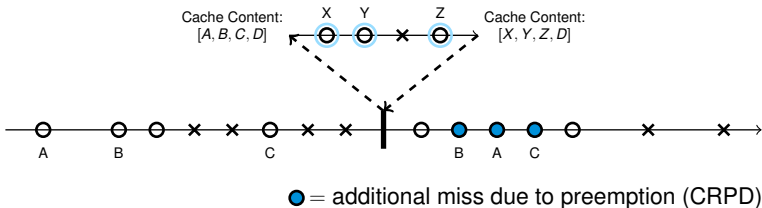
BRT = Block Reload Time

# 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.*

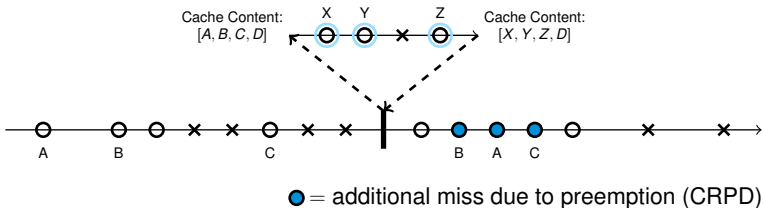


# Evicting Cache Blocks

[?]

## 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.*



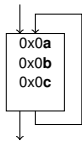
$$\text{CRPD}_{\text{ECB}}^s = \begin{cases} 0 & \text{if } \text{ECB}(s) = \emptyset \\ \text{BRT} \times n & \text{otherwise} \end{cases}$$

# Impact of the preempting task on the preempted task

## CRPD (using UCB and ECB)

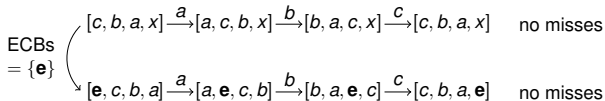
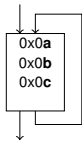
$$CRPD_{UCB\&ECB} = \sum_{s=1}^c \min(CRPD_{UCB}^s, CRPD_{ECB}^s)$$

# Impact of the preempting task on the preempted task: Example



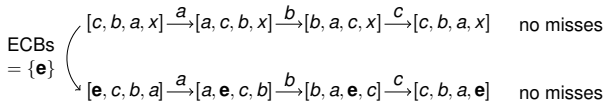
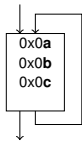
$[c, b, a, x] \xrightarrow{a} [a, c, b, x] \xrightarrow{b} [b, a, c, x] \xrightarrow{c} [c, b, a, x]$  no misses

# Impact of the preempting task on the preempted task: Example



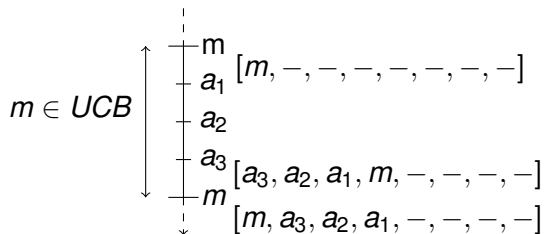
- $\text{CRPD}_{\text{UCB}} \Rightarrow |\text{UCB}| = 3$
- $\text{CRPD}_{\text{ECB}} \Rightarrow n = 4$
- $\text{CRPD}_{\text{UCB\&ECB}} = \min(\text{CRPD}_{\text{UCB}}, \text{CRPD}_{\text{ECB}}) \Rightarrow 3$ 
  - ▶ **Overestimation: number of additional misses = 0 < 3**

# Impact of the preempting task on the preempted task: Example



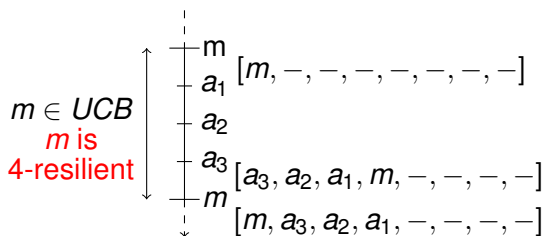
- $\text{CRPD}_{\text{UCB}} \Rightarrow |\text{UCB}| = 3$
- $\text{CRPD}_{\text{ECB}} \Rightarrow n = 4$
- $\text{CRPD}_{\text{UCB\&ECB}} = \min(\text{CRPD}_{\text{UCB}}, \text{CRPD}_{\text{ECB}}) \Rightarrow 3$ 
  - ▶ Overestimation: number of additional misses =  $0 < 3$
- Why?
  - ▶  $|\text{ECB}|$  to evict a UCB = 2, but
  - ▶  $|\text{ECB}| = 1$
  - ▶ A single ECB is not sufficient to evict a UCB.

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





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



## Definition (I-Resilience)

*A memory block  $m$  is called  $I$ -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  $I$  accesses.*

# Resilience analysis

## Definition (I-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.*

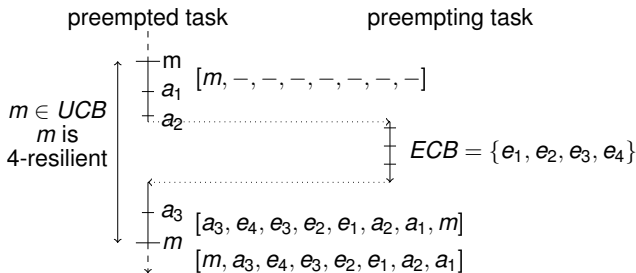
- No UCB is  $n$ -resilient, i.e., no UCB remains useful after a preemption with  $n$  ECBs.
- Each  $(l + 1)$ -resilient UCB is also  $l$ -resilient.
- Each UCB is at least 0-resilient.

# Resilience analysis

## Definition (I-Resilience)

A memory block  $m$  is called  $I$ -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  $I$  accesses.

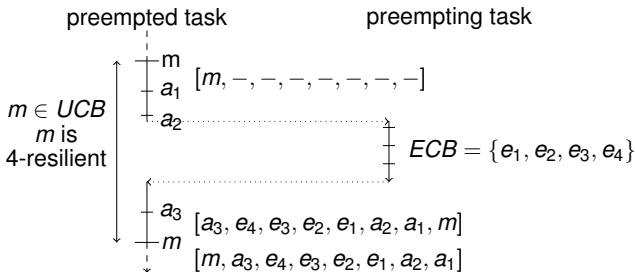


# Resilience analysis

## Definition (I-Resilience)

A memory block  $m$  is called  $I$ -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  $I$  accesses.



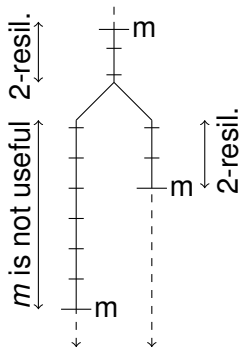
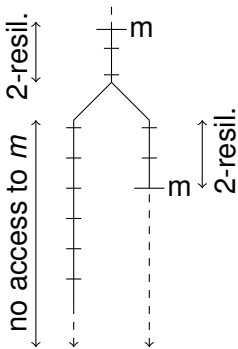
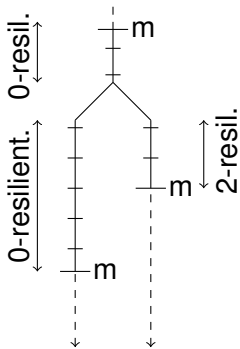
In general: if  $|ECB| \leq I$  then the UCB is not evicted

# Resilience analysis

## Definition (I-Resilience)

A memory block  $m$  is called  $I$ -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  $I$  accesses.



## CRPD (Combining UCB and ECB by using Resilience)

$$\underbrace{|UCB|}_{\text{useful}} \setminus \underbrace{\{m \mid m \text{ is } |ECB|\text{-resilient}\}}_{\text{remain useful}}$$

*blocks contributing to CRPD*

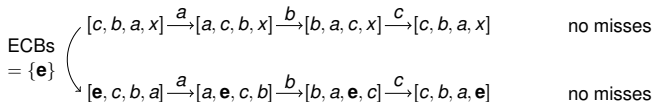
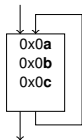
## CRPD (Combining UCB and ECB by using Resilience)

$$CRPD \leq BRT \times \left| \underbrace{UCB}_{\text{useful}} \setminus \underbrace{\{m \mid m \text{ is } |ECB|\text{-resilient}\}}_{\text{remain useful}} \right|$$

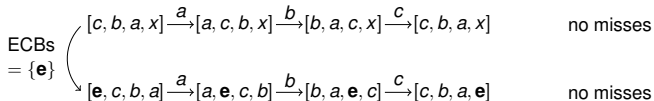
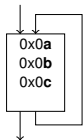
*blocks contributing to CRPD*



# Bounding the CRPD using Resilience: Example

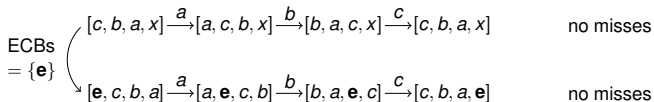
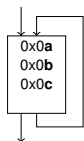


# Bounding the CRPD using Resilience: Example



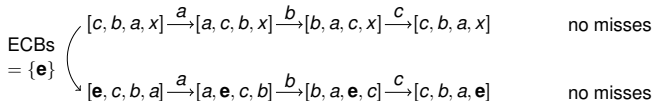
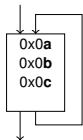
- - ▶  $|\text{ECB}| = 1$
  - ▶  $a, b$  and  $c$  are 1-resilient
  - ▶  $\text{CRPD}_{\text{UCB\&ECB}}^{\text{res}} = \text{BRT} \times |\text{UCB} \setminus \{m \mid m \text{ is } |\text{ECB}|\text{-resilient}\}| = 0$

# Bounding the CRPD using Resilience: Example



- - ▶  $|\text{ECB}| = 1$
  - ▶  $a, b$  and  $c$  are 1-resilient
  - ▶  $\text{CRPD}_{\text{UCB\&ECB}}^{\text{res}} = \text{BRT} \times |\text{UCB} \setminus \{m \mid m \text{ is } |\text{ECB}|\text{-resilient}\}| = 0$
- Instead of:  $\text{CRPD}_{\text{UCB\&ECB}} = \min(\text{CRPD}_{\text{UCB}}, \text{CRPD}_{\text{ECB}}) = 3 \times \text{BRT}$

# Bounding the CRPD using Resilience: Example



- - ▶  $|ECB| = 1$
  - ▶  $a, b$  and  $c$  are 1-resilient
  - ▶  $CRPD_{UCB\&ECB}^{res} = BRT \times |UCB \setminus \{m \mid m \text{ is } |ECB|\text{-resilient}\}| = 0$
- Instead of:  $CRPD_{UCB\&ECB} = \min(CRPD_{UCB}, CRPD_{ECB}) = 3 \times BRT$

- 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



Altmeyer, S. and Burguière, C. (2009).

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

In ECRTS '09 pp. 109–118, IEEE Computer Society.



Lee, C.-G., Hahn, J., Min, S. L., Ha, R., Hong, S., Park, C. Y., Lee, M. and Kim, C. S. (1996).

Analysis of cache-related preemption delay in fixed-priority preemptive scheduling.

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



Negi, H. S., Mitra, T. and Roychoudhury, A. (2003).

Accurate estimation of cache-related preemption delay.

In CODES+ISSS'03 ACM.



Reineke, J. (2008).

Caches in WCET Analysis.

PhD thesis, Universität des Saarlandes, Saarbrücken.



Schneider, J. (2000).

Cache and pipeline sensitive fixed priority scheduling for preemptive real-time systems.

In In Proceedings of the 21st IEEE Real-Time Systems Symposium (RTSS'2000) pp. 195–204,.



Staschulat, J. and Ernst, R. (2007).

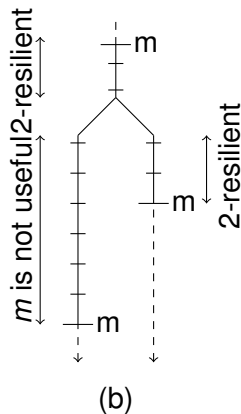
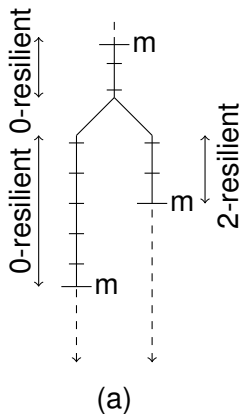
Scalable precision cache analysis for real-time software.

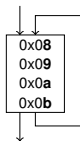
ACM TECS 6, 25.



Tan, Y. and Mooney, V. (2004).

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





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

- $|\text{UCB}(s)| = 4$
- $|\text{ECB}(s)| = 1$
- $n = 4$
- number of additional misses = 4



- using UCB [?]:

$$\text{CRPD}_{\text{UCB}} = \text{BRT} \cdot |\{s_i \mid \exists m \in \text{UCB} : m \bmod c = s_i\}|$$

---

- using ECB [?]:

$$\text{CRPD}_{\text{ECB}} = \text{BRT} \cdot |\{s_i \mid \exists m \in \text{ECB} : m \bmod c = s_i\}|$$

---

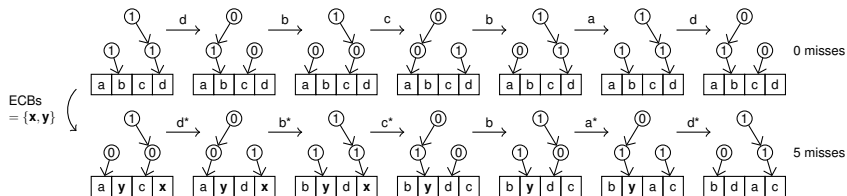
- using UCB and ECB [?, ?]:

$$\text{CRPD}_{\text{UCB\&ECB}} = \text{BRT} \cdot |\{s_i \mid \begin{array}{l} \exists m \in \text{UCB} : m \bmod c = s_i \\ \wedge \exists m' \in \text{ECB} : m' \bmod c = s_i \end{array}\}|$$

$$\begin{array}{l} \text{ECBs} \\ = \{\mathbf{x}\} \end{array} \left( \begin{array}{l} [b, a] \xrightarrow{a} [b, a] \xrightarrow{e^*} [e, b] \xrightarrow{b} [e, b] \xrightarrow{c^*} [c, e] \xrightarrow{e} [c, e] \quad 2 \text{ misses} \\ [x, b] \xrightarrow{a^*} [a, x] \xrightarrow{e^*} [e, a] \xrightarrow{b^*} [b, e] \xrightarrow{c^*} [c, b] \xrightarrow{e^*} [e, c] \quad 5 \text{ misses} \end{array} \right.$$

$$\begin{array}{l} \text{ECBs} \\ = \{\mathbf{x}\} \end{array} \left( \begin{array}{l} [b, a] \xrightarrow{a} [b, a] \xrightarrow{e^*} [e, b] \xrightarrow{b} [e, b] \xrightarrow{c^*} [c, e] \xrightarrow{e} [c, e] \quad 2 \text{ misses} \\ [x, b] \xrightarrow{a^*} [a, x] \xrightarrow{e^*} [e, a] \xrightarrow{b^*} [b, e] \xrightarrow{c^*} [c, b] \xrightarrow{e^*} [e, c] \quad 5 \text{ misses} \end{array} \right)$$

- $|\text{UCB}(s)| = 2$
- $|\text{ECB}(s)| = 1$
- $n = 2$
- **But: number of additional misses = 3**



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