Cache Related Preemption Delay for Set-Associative Caches

Resilience Analysis

Sebastian Altmeyer, Claire Burguière, Jan Reineke

AVACS Workshop, Oldenburg 2009

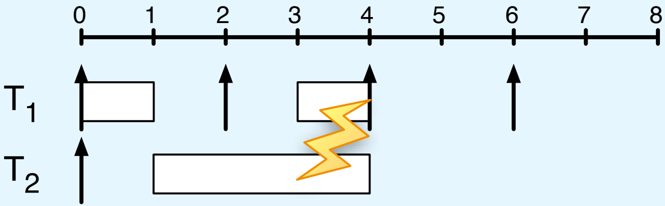
SAARLAND UNIVERSITY
COMPUTER SCIENCE

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 $C_1 = 1, C_2 = 3$.

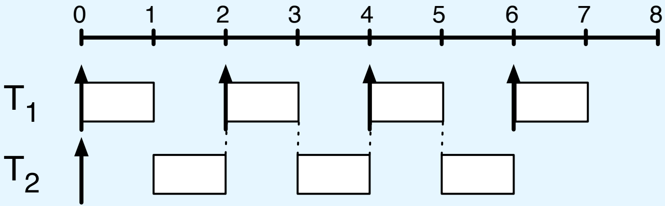


Why use preemptive scheduling?

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

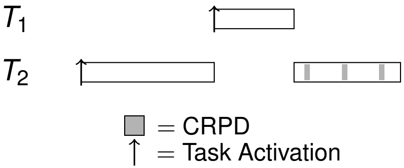
Example

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).



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

  $\Rightarrow$ Some UCBs remain useful under preemption

Useful Cache Block - [?]

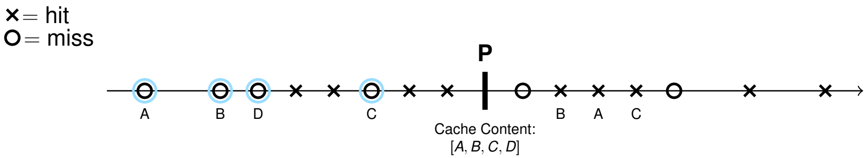
SAARLAND UNIVERSITY
COMPUTER SCIENCE

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 - [?]

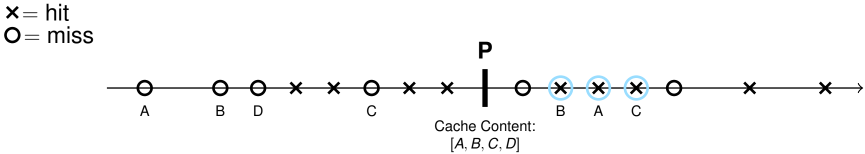
SAARLAND UNIVERSITY
COMPUTER SCIENCE

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.*



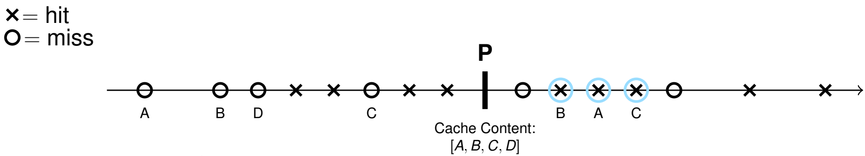
Useful Cache Block - [?]

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.*



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

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

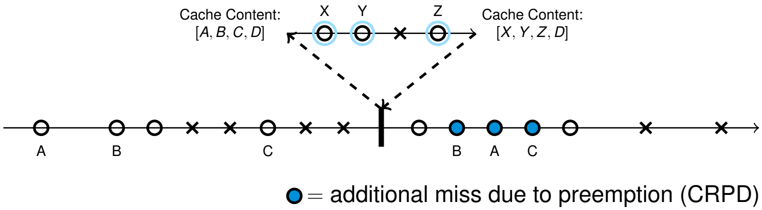
n = associativity

BRT = Block Reload Time

Evicting Cache Blocks [?]

Definition (Evicting Cache Blocks (ECB))

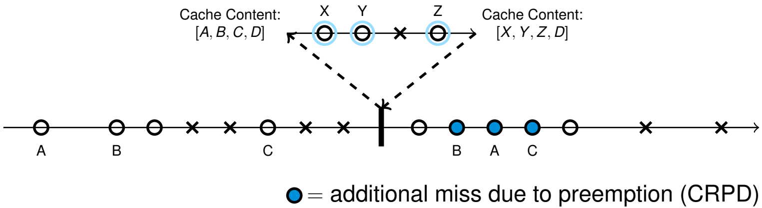
A memory block of the preempting task is called an evicting cache block, if it may be accessed during the execution of the preempting task.



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.



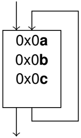
$$\mathrm{CRPD}^{s}_{\mathrm{ECB}} = \begin{cases} 0 & \text{if } \mathrm{ECB}(s) = \emptyset \\ \mathrm{BRT} \times n & \text{otherwise} \end{cases}$$

Impact of the preempting task on the preempted task

CRPD (using UCB and ECB)

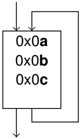
$$CRPD_{UCB\&ECB} = \sum_{s=1}^{c} min(CRPD^s_{UCB}, CRPD^s_{ECB})$$

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



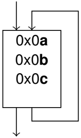
ECBs $= \{\mathbf{e}\}$

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

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

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

Impact of the preempting task on the preempted task: Example



ECBs $= \{\mathbf{e}\}$

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

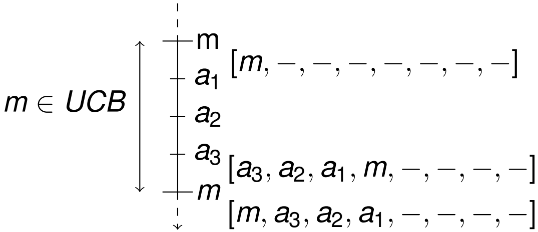
$$[\mathbf{e},c,b,a] \xrightarrow{a} [a,\mathbf{e},c,b] \xrightarrow{b} [b,a,\mathbf{e},c] \xrightarrow{c} [c,b,a,\mathbf{e}] \quad \text{no misses}$$

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

- Why?
  - $|\mathrm{ECB}|$ to evict a UCB $= 2$, but
  - $|\mathrm{ECB}| = 1$
  - A single ECB is not sufficient to evict a UCB.

SAARLAND UNIVERSITY COMPUTER SCIENCE

Determining *max*|ECB|, such that no additional cache miss occur



Combining UCB and ECB: Refinement

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

$m \in UCB$

m is 4-resilient

m $[m, -, -, -, -, -, -, -]$

a_1

a_2

a_3 $[a_3, a_2, a_1, m, -, -, -, -]$

m $[m, a_3, a_2, a_1, -, -, -, -]$

Resilience analysis

SAARLAND UNIVERSITY
COMPUTER SCIENCE

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

SAARLAND UNIVERSITY
COMPUTER SCIENCE

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.

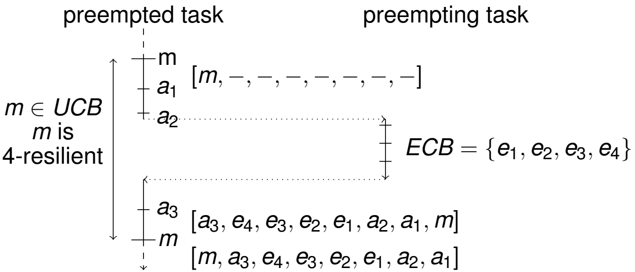
Resilience analysis

SAARLAND UNIVERSITY
COMPUTER SCIENCE

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.*

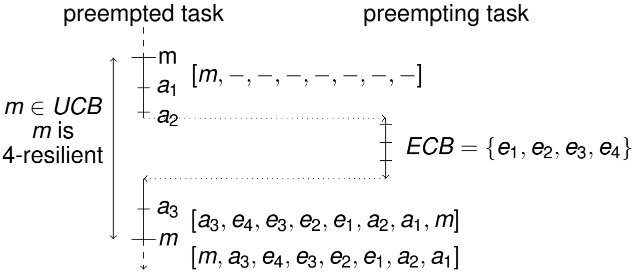


Resilience analysis

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.*



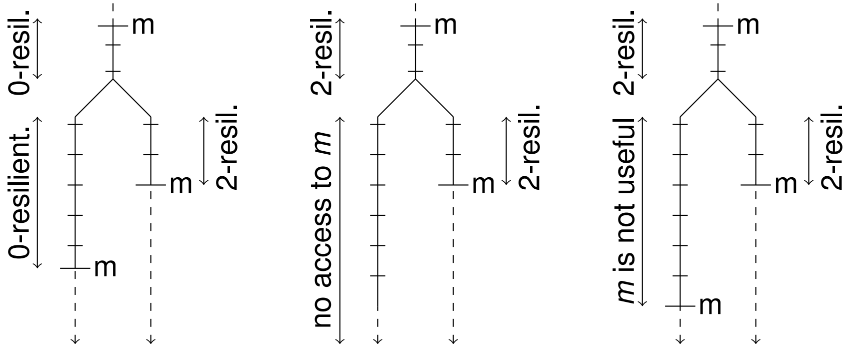
In general: if $|\text{ECB}| \leq l$ then the UCB is not evicted

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.*



Bounding the CRPD using Resilience

CRPD (Combining UCB and ECB by using Resilience)

$$\left|\overbrace{\underbrace{UCB}_{\textit{useful}} \setminus \underbrace{\{m \mid m \textit{ is } |ECB|\textit{-resilient}\}}_{\textit{remain useful}}}^{\textit{blocks contributing to CRPD}}\right|$$

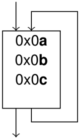
Bounding the CRPD using Resilience

SAARLAND UNIVERSITY
COMPUTER SCIENCE

CRPD (Combining UCB and ECB by using Resilience)

$$CRPD \leq BRT \times |\overbrace{\underbrace{UCB}_{\textit{useful}} \setminus \underbrace{\{m \mid m \textit{ is } |ECB|\textit{-resilient}\}}_{\textit{remain useful}}}^{\textit{blocks contributing to CRPD}}|$$

Bounding the CRPD using Resilience: Example

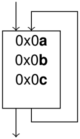


ECBs $= \{\mathbf{e}\}$

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

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

Bounding the CRPD using Resilience: Example



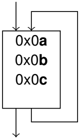
ECBs $= \{\mathbf{e}\}$

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

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

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

Bounding the CRPD using Resilience: Example



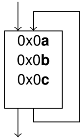
ECBs $= \{\mathbf{e}\}$

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

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

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

Bounding the CRPD using Resilience: Example



ECBs $= \{\mathbf{e}\}$

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

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

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

Conclusions

SAARLAND UNIVERSITY
COMPUTER SCIENCE

- 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.

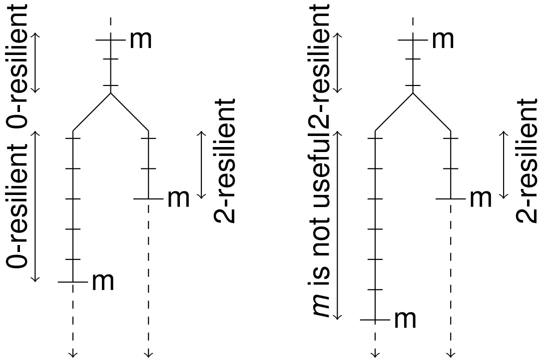
Reineke, J. (2008).
Caches in WCET Analysis.
PhD thesis, Universität des Saarlandes, Saarbrücken.

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.

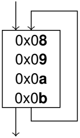
Tan, Y. and Mooney, V. (2004).
Integrated intra- and inter-task cache analysis for preemptive multi-tasking real-time

l-resilience analysis



(a) (b)

CPRD using ECB: Pitfall



ECBs $= \{\mathbf{e}\}$

$$[b, a, 9, 8] \xrightarrow{8} [8, b, a, 9] \xrightarrow{9} [9, 8, b, a] \xrightarrow{a} [a, 9, 8, b] \xrightarrow{b} [b, a, 9, 8] \quad \text{0 misses}$$

$$[\mathbf{e}, b, a, 9] \xrightarrow{8^*} [8, \mathbf{e}, b, a] \xrightarrow{9^*} [9, 8, \mathbf{e}, b] \xrightarrow{a^*} [a, 9, 8, \mathbf{e}] \xrightarrow{b^*} [b, a, 9, 8] \quad \text{4 misses}$$

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

Upper-bound on the CRPD - direct-mapped caches

- using UCB [?]:

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

- using ECB [?]:

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

- using UCB and ECB [?, ?]:

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

CRPD for FIFO: Pitfalls

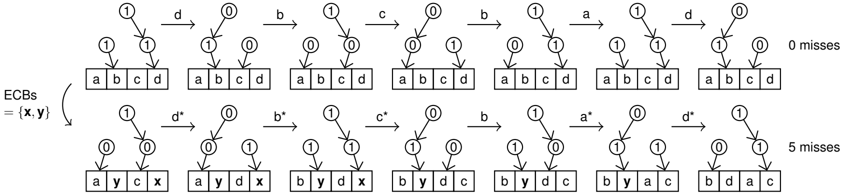
$$
\text{ECBs} = \{\mathbf{x}\} \begin{cases} [b, a] \xrightarrow{a} [b, a] \xrightarrow{e^*} [e, b] \xrightarrow{b} [e, b] \xrightarrow{c^*} [c, e] \xrightarrow{e} [c, e] & \text{2 misses} \\ [\mathbf{x}, b] \xrightarrow{a^*} [a, \mathbf{x}] \xrightarrow{e^*} [e, a] \xrightarrow{b^*} [b, e] \xrightarrow{c^*} [c, b] \xrightarrow{e^*} [e, c] & \text{5 misses} \end{cases}
$$

CRPD for FIFO: Pitfalls

$$\begin{array}{l} \text{ECBs} \\ = \{\mathbf{x}\} \end{array} \begin{cases} [b, a] \xrightarrow{a} [b, a] \xrightarrow{e^*} [e, b] \xrightarrow{b} [e, b] \xrightarrow{c^*} [c, e] \xrightarrow{e} [c, e] & \text{2 misses} \\ [\mathbf{x}, b] \xrightarrow{a^*} [a, \mathbf{x}] \xrightarrow{e^*} [e, a] \xrightarrow{b^*} [b, e] \xrightarrow{c^*} [c, b] \xrightarrow{e^*} [e, c] & \text{5 misses} \end{cases}$$

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

CRPD for PLRU: Pitfalls



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