Cache related pre-emption delay aware response time analysis for fixed priority pre-emptive systems

Sebastian Altmeyer, Robert I. Davis, Claire Maiza

RTSS 2011, Vienna

SAARLAND UNIVERSITY
COMPUTER SCIENCE

RTS York

Verimag

1. Cache Related Pre-emption Delay
   - Useful Cache Blocks
   - Evicting Cache Blocks

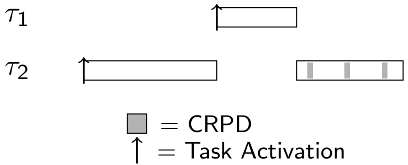
2. Response Time Analysis
   - Analysis/Review of existing Approaches
   - New Approach: ECB Union
   - Correct Handling of Blocking Time

3. Evaluation
   - Case Study
   - Generated Test Cases

4. Conclusions & Future Work

Pre-emptively Scheduled Systems: Cache Related Pre-emption Delay

- Pre-emptive scheduling
- Cache related pre-emption delay (CRPD):
  - Impact of pre-emption on the cache content
  - Overall cost of additional reloads due to pre-emption

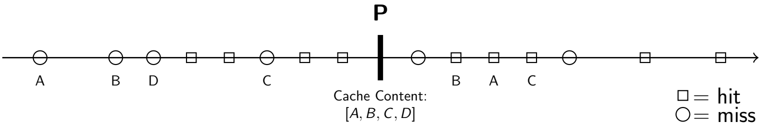


Useful Cache Blocks

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

a) m may be cached at P

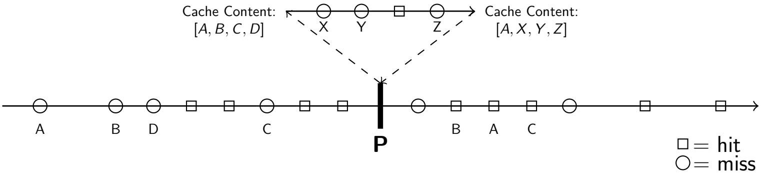
b) m may be reused at program point P' that may be reached from P with no eviction of m on this path.



$$UCB = \{A, B, C\}$$

Evicting Cache Blocks

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



$$ECB = \{X, Y, Z\}$$

Cache Related Pre-emption Delay - Notation

Presentation restricted to direct-mapped caches only:

Sets of ECBs and UCBs are sets of integers:

$$s \in \text{UCB}_i \Leftrightarrow \tau_i \text{ has a useful cache block in cache-set } s$$

$$s \in \text{ECB}_i \Leftrightarrow \tau_i \text{ may evict a cache block in cache-set } s$$

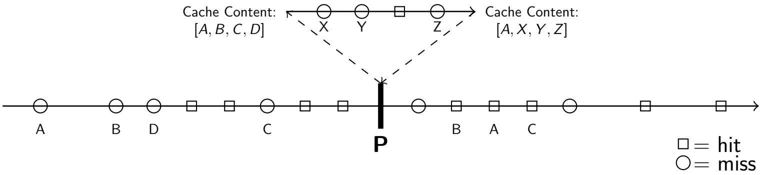
pre-emption cost task τ_j pre-empting τ_i (BRT block reload time):

$$\text{BRT} \cdot |\text{UCB}_i \cap \text{ECB}_j|$$

Cache Related Pre-emption Delay - Example

τ_1 pre-empts τ_2

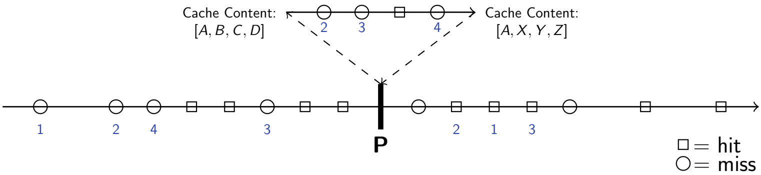
$$\mathrm{ECB}_1 = \{X, Y, Z\} \ \mathrm{UCB}_2 = \{A, B, C\}$$



Cache Related Pre-emption Delay - Example

τ_1 pre-empts τ_2

$$\text{ECB}_1 = \{2,3,4\} \quad \text{UCB}_2 = \{1,2,3\}$$



$$CRPD_{1,2} = \text{BRT} \cdot |\text{UCB}_2 \cap \text{ECB}_1|$$
$$= \text{BRT} \cdot |\{1,2,3\} \cap \{2,3,4\}| = \text{BRT} \cdot |\{2,3\}|$$

UCBs in cache-set 2 and 3 may be evicted

$\Rightarrow$ 2 pre-emption misses

Response Time Analysis (for fixed priorities)

$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j)$$

Response Time $R_i =$ finishing time $-$ activation time
no deadline miss $\Leftrightarrow \forall \tau_i : R_i \leq D_i$

(exec. time C_i, period T_i, deadline D_i, tasks with higher priority $\mathrm{hp}(i)$)

Response Time Analysis (for fixed priorities)

$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

$\gamma_{i,j}$ denotes the pre-emption cost

Response Time R_i = finishing time − activation time
no deadline miss $\Leftrightarrow \forall \tau_i : R_i \leq D_i$

(exec. time C_i, period T_i, deadline D_i, tasks with higher priority $\mathrm{hp}(i)$)

Response Time Analysis with CRPD

$$R_i = C_i + \sum_{\forall j \in \mathsf{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

$\gamma_{i,j}$ denotes the pre-emption cost

But what is the precise meaning of γ?

UCB only or ECB only (Busquets-Mataix et al., Lee et al.)

$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

$$\gamma_{i,j}^{\mathrm{ecb}} = \mathrm{BRT} \cdot |\mathrm{ECB}_j| \quad \text{or} \quad \gamma_{i,j}^{\mathrm{ucb}} = \mathrm{BRT} \cdot \max_{\forall k \in \mathrm{aff}(i,j)} \{|\mathrm{UCB}_k|\}$$

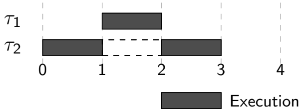
$$(\mathrm{aff}(i,j) = \mathrm{hep}(i) \cap \mathrm{lp}(j))$$

UCB only or ECB only (Busquets-Mataix et al., Lee et al.)

$$R_i = C_i + \sum_{\forall j \in \text{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

$$\gamma_{i,j}^{\text{ecb}} = \text{BRT} \cdot |\text{ECB}_j| \quad \text{or} \quad \gamma_{i,j}^{\text{ucb}} = \text{BRT} \cdot \max_{\forall k \in \text{aff}(i,j)} \{|\text{UCB}_k|\}$$

$$(\text{aff}(i,j) = \text{hep}(i) \cap \text{lp}(j))$$



	UCB_i	ECB_i
τ_1	$\{1,2\}$	$\{1,2\}$
τ_2	$\{3,4\}$	$\{1,2,3,4\}$

$$\gamma_{2,1}^{\text{ucb}} = \gamma_{2,1}^{\text{ecb}} = 2$$

$$\text{actual cost} = 0$$

Why not use a simple combination?

$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

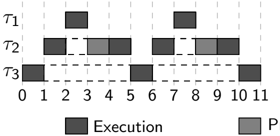
$$\gamma_{i,j} = \mathrm{BRT} \cdot |\mathrm{UCB}_i \cap \mathrm{ECB}_j|$$

Why not use a simple combination?

$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

$$\gamma_{i,j} = \mathrm{BRT} \cdot |\mathrm{UCB}_i \cap \mathrm{ECB}_j|$$

$$C_1 = 1,\ C_2 = 2,\ C_3 = 3,\ \mathrm{BRT} = 1$$



	UCB_i	ECB_i
τ_1	$\emptyset$	$\{1, 2\}$
τ_2	$\{1\}$	$\{1, 2\}$
τ_3	$\{3, 4\}$	$\{1, 2, 3, 4\}$

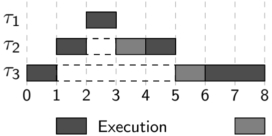
τ_1 pre-empting τ_2 causes higher costs (1) than τ_1 pre-empting τ_3 (0)

Why not use a simple combination?

$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

$$\gamma_{i,j} = \mathrm{BRT} \cdot |\mathrm{UCB}_i \cap \mathrm{ECB}_j|$$

$$C_1 = 1,\ C_2 = 2,\ C_3 = 3,\ \mathrm{BRT} = 1$$



	UCB_i	ECB_i
τ_1	$\emptyset$	$\{2,3\}$
τ_2	$\{1,2\}$	$\{1,2\}$
τ_3	$\{3,4\}$	$\{1,2,3,4\}$

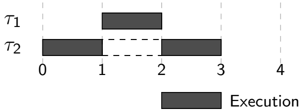
Nested pre-emption causes higher costs (2) than any non-nested (1)

UCB Union (Tan et al.)

$$\gamma_{i,j}^{\text{tan}} = \text{BRT} \cdot \underbrace{\left| \overbrace{\left(\bigcup_{\forall k \in \text{aff}(i,j)} \text{UCB}_k \right)}^{\textit{all possibly affected UCBs}} \cap \text{ECB}_j \right|}_{\textit{that are evicted by task } \tau_j}$$

UCB Union (Tan et al.)

$$\gamma_{i,j}^{\text{tan}} = \text{BRT} \cdot \underbrace{\left| \overbrace{\left(\bigcup_{\forall k \in \text{aff}(i,j)} \text{UCB}_k \right)}^{\textit{all possibly affected UCBs}} \cap \text{ECB}_j \right|}_{\textit{that are evicted by task } \tau_j}$$



	UCB_i	ECB_i
τ_1	$\{1,2\}$	$\{1,2\}$
τ_2	$\{3,4\}$	$\{1,2,3,4\}$

$$\gamma_{2,1}^{\text{ucb}} = \gamma_{2,1}^{\text{ecb}} = 2$$

$$\gamma_{2,1}^{\text{tan}} = 0$$

UCB Union (Tan et al.)

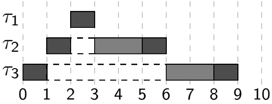
$$\gamma_{i,j}^{\text{tan}} = \text{BRT} \cdot \underbrace{\left| \overbrace{\left(\bigcup_{\forall k \in \text{aff}(i,j)} \text{UCB}_k \right)}^{\textit{all possibly affected UCBs}} \cap \text{ECB}_j \right|}_{\textit{that are evicted by task } \tau_j}$$

- safe combination of ECBs and UCBs
- dominates ECB-Only ($\gamma_{i,j}^{\text{ecb}} = \text{BRT} \cdot |\text{ECB}_j|$)

UCB Union (Tan et al.)

$$\gamma_{i,j}^{\text{tan}} = \text{BRT} \cdot \left| \underbrace{\overbrace{\left(\bigcup_{\forall k \in \text{aff}(i,j)} \text{UCB}_k \right)}^{\textit{all possibly affected UCBs}} \cap \text{ECB}_j}_{\textit{that are evicted by task } \tau_j} \right|$$

$$C_1 = 1,\ C_2 = C_3 = 2,\ \text{BRT} = 1$$



	UCB_i	ECB_i
τ_1	$\emptyset$	$\{1,2,3,4\}$
τ_2	$\{1,2\}$	$\{1,2,3,4\}$
τ_3	$\{3,4\}$	$\{1,2,3,4\}$

$$\gamma_{3,1}^{\text{tan}} = 4 \wedge \gamma_{3,2}^{\text{tan}} = 2 \rightarrow \text{total pre-emption cost} = 6$$

$$\text{actual cost} = 4$$

New Approach: ECB Union

$$\gamma_{i,j}^{\text{new}} = \text{BRT} \cdot \underbrace{\max_{\forall k \in \text{aff}(i,j)} \left\{ \left| \text{UCB}_k \cap \overbrace{\left(\bigcup_{h \in \text{hp}(j) \cup \{j\}} \text{ECB}_h \right)}^{\textit{impact of all higher priority tasks}} \right| \right\}}_{\textit{affected task with highest CRPD}}$$

New Approach: ECB Union

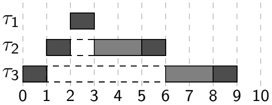
$$\gamma_{i,j}^{\text{new}} = \text{BRT} \cdot \underbrace{\max_{\forall k \in \text{aff}(i,j)} \left\{ \left| \text{UCB}_k \cap \overbrace{\left(\bigcup_{h \in \text{hp}(j) \cup \{j\}} \text{ECB}_h \right)}^{\textit{impact of all higher priority tasks}} \right| \right\}}_{\textit{affected task with highest CRPD}}$$

- safe combination of ECBs and UCBs
- dominates UCB-Only ($\gamma_{i,j}^{\text{ucb}} = \text{BRT} \cdot \max_{\forall k \in \text{aff}(i,j)} \{|\text{UCB}_k|\}$)

New Approach: ECB Union

$$\gamma_{i,j}^{\text{new}} = \text{BRT} \cdot \underbrace{\max_{\forall k \in \text{aff}(i,j)} \left\{ \left| \text{UCB}_k \cap \overbrace{\left(\bigcup_{h \in \text{hp}(j) \cup \{j\}} \text{ECB}_h \right)}^{\textit{impact of all higher priority tasks}} \right| \right\}}_{\textit{affected task with highest CRPD}}$$

$$C_1 = 1,\ C_2 = C_3 = 2,\ \text{BRT} = 1$$



	UCB_i	ECB_i
τ_1	$\emptyset$	$\{1,2,3,4\}$
τ_2	$\{1,2\}$	$\{1,2,3,4\}$
τ_3	$\{3,4\}$	$\{1,2,3,4\}$

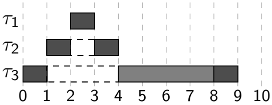
$$\gamma_{3,1}^{\text{tan}} = 4 \wedge \gamma_{3,2}^{\text{tan}} = 2 \Rightarrow 6 \qquad \gamma_{3,1}^{\text{new}} = 2 \wedge \gamma_{3,2}^{\text{new}} = 2 \Rightarrow 4$$

actual cost $= 4$

New Approach: ECB Union

$$\gamma_{i,j}^{\text{new}} = \text{BRT} \cdot \underbrace{\max_{\forall k \in \text{aff}(i,j)} \left\{ \left| \text{UCB}_k \cap \overbrace{\left(\bigcup_{h \in \text{hp}(j) \cup \{j\}} \text{ECB}_h \right)}^{\textit{impact of all higher priority tasks}} \right| \right\}}_{\textit{affected task with highest CRPD}}$$

$$C_1 = 1,\ C_2 = C_3 = 2,\ \text{BRT} = 1$$

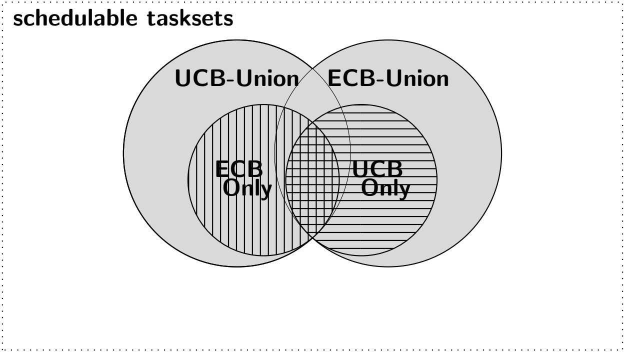


	UCB_i	ECB_i
τ_1	$\emptyset$	$\{1,2\}$
τ_2	$\emptyset$	$\{3,4\}$
τ_3	$\{1,2,3,4\}$	$\{1,2,3,4\}$

$$\gamma_{3,1}^{\text{tan}} = 2 \wedge \gamma_{3,2}^{\text{tan}} = 2 \Rightarrow 4 \qquad \gamma_{3,1}^{\text{new}} = 2 \wedge \gamma_{3,2}^{\text{new}} = 4 \Rightarrow 6$$

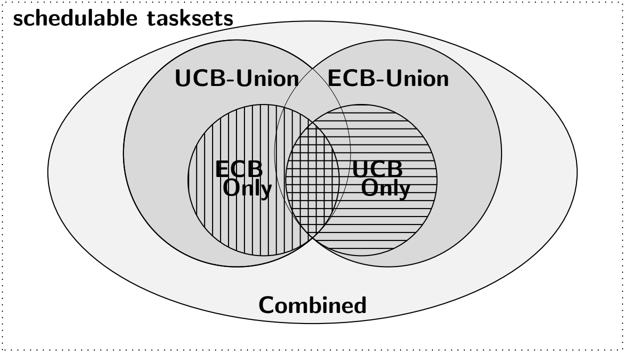
$$\text{actual cost} = 4$$

Combined Approach



The larger the area, the more tasksets deemed schedulable.

Combined Approach

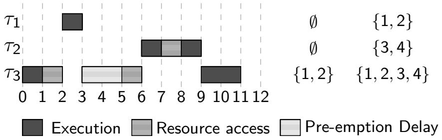


The larger the area, the more tasksets deemed schedulable.

$$R_i^{\text{comb}} = \min(R_i^{\text{tan}}, R_i^{\text{new}})$$

Blocking Time (Stack Resource Protocol)

$$R_i = C_i + B_i + \sum_{\forall j \in \mathbf{hp}(i)} \left\lceil \frac{R_i + J_j}{T_j} \right\rceil (C_j + \gamma_{i,j})$$



	UCB_i	ECB_i
τ_1	$\emptyset$	$\{1, 2\}$
τ_2	$\emptyset$	$\{3, 4\}$
τ_3	$\{1, 2\}$	$\{1, 2, 3, 4\}$

τ_2 and τ_3 share a common resource

Task τ_2 can be blocked by execution of τ_3 and pre-emption delay (τ_1 pre-empting τ_3)

ECB-Only accounts for this implicitly
all others must be extended (see paper)

Evaluation

1. case study (benchmarks from Mälardalen benchmark suite)

2. randomly generated test cases

Case Study – Benchmarks

	WCET	UCBs	ECBs
bs	445	5	35
minmax	504	9	79
fac	1,252	4	24
fibcall	1,351	5	24
insertsort	6,573	10	41
loop3	13,449	4	817
select	17,088	15	151
qsort-exam	22,146	15	170
fir	29,160	9	105
sqrt	39,962	14	477
ns	43,319	13	64
qurt	214,076	14	484
crc	290,782	14	144
matmult	742,585	23	100
bsort100	1,567,222	35	62

Periods: $\forall_i : T_i = c \cdot C_i$; c varied from 15 upwards $\Rightarrow$ utilization from 1.0 downwards

(ARM7, direct-mapped instruction, cache size 2kB, line size 8 Bytes (256 cache sets) and BRT $= 8\mu s$)

Case Study – Results

breakdown utilization: max utilization s.t. taskset was deemed schedulable

Analysis	Breakdown utilization:
No Pre-emption Cost	0.95
Combined	0.767
ECB-Union	0.767
UCB-Only	0.75
UCB-Union	0.698
ECB-Only	0.612

Generated Test Cases – Setting

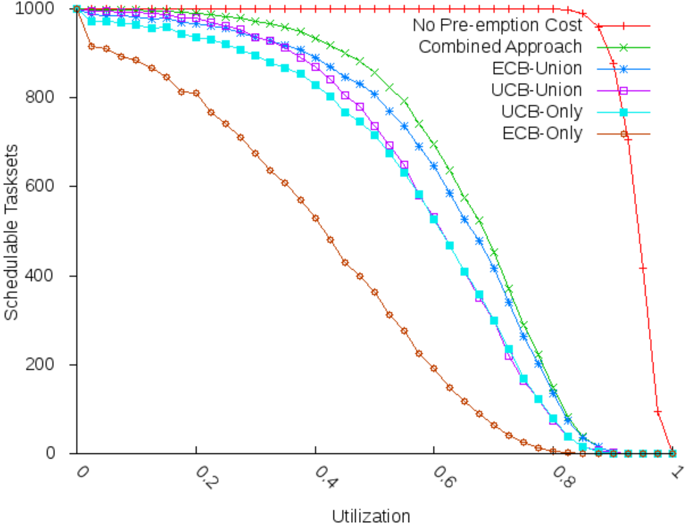
Task set:

- 10 tasks
- periods T_i range from 5ms to 500ms (log-uniform distribution)
- task utilization U_i generated using UUnifast
- execution times $C_i = U_i \cdot T_i$
- implicit deadlines, priorities in deadline monotonic order

Pre-emption costs:

- number of cache sets ($CS = 256$)
- block-reload time ($BRT = 8\mu s$)
- cache usage using UUnifast ($CU = 10$)
- reuse factor (UCBs), uniform distribution $[0; |\mathrm{ECB}|]$

Generated Test Cases – Results



Why does ECB-Union perform better than UCB-Union?

- UCB-Union overapproximates evicted UCBs
- ECB-Union overapproximates evicted ECBs
- always more ECBs than UCBs
- also UCB-Only better than ECB-Only

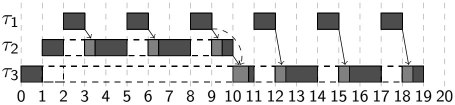
holds even for different parameter settings
(see evaluation in paper)

Conclusions

- Analysis of Response Time Analyses with CRPD
- New Approaches (ECB-Union and Combined)
- Corrected Handling of Blocking Time
- Thorough Evaluation (Case Study; generated test cases with varying parameters)

Future Work

- ECB Union and UCB Union still pessimistic:



ECB Union assumes task τ_1 pre-empts τ_2 up to six times
but, task τ_1 pre-empts τ_2 at most three times

- Pre-emption Cost and Fixed Priority FIFO Scheduling
- Influence of the task mapping on CRPD
- Comparison with ScratchPad Memories

Thanks for your attention.

Questions?