Analysis of Preemptively Scheduled Hard Real-Time Systems

Sebastian Altmeyer

October 25th, 2012

SAARLAND UNIVERSITY
COMPUTER SCIENCE

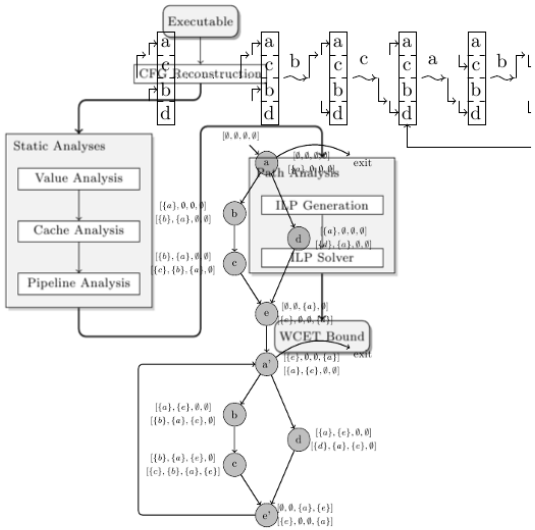
Hard Real-Time Systems

- embedded systems with stringent timing constraints (deadlines)
- missing a deadline is a complete system failure
- examples are airbags, anti-lock breaking system (ABS), flight control systems etc.

$$\Rightarrow$$

Correct timing behavior must be guaranteed

Timing Analysis

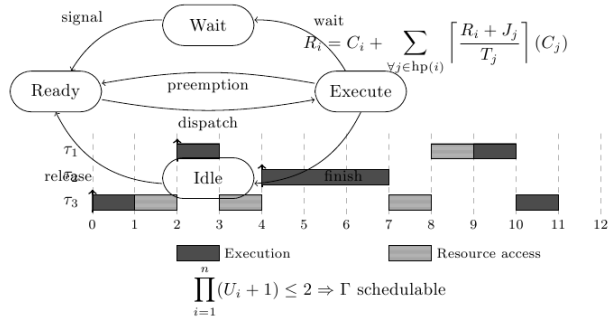


Computes execution time bounds of all tasks

$\Longrightarrow$

WCET

Scheduling Analysis



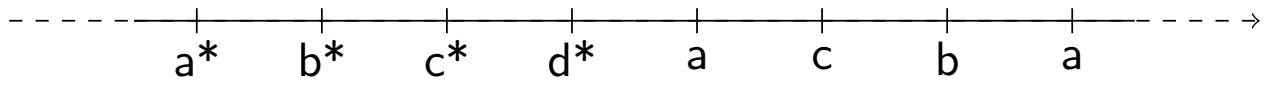
$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i + J_j}{T_j} \right\rceil (C_j)$$

$$\prod_{i=1}^{n} (U_i + 1) \leq 2 \Rightarrow \Gamma \text{ schedulable}$$

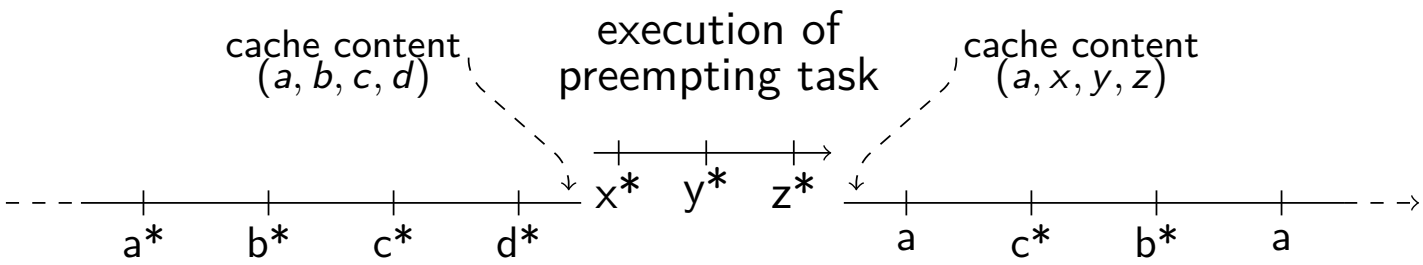
Checks that no task misses its deadline

Traditionally: execution time bounds are the only interface

Still valid in Presence of Caches and Preemption?



(non-preempted execution[1])



(preempted execution)

- Cache-related preemption delay (CRPD)
  - denotes overall cost of additional reloads due to preemption
  - may increase execution time significantly.

$$\Rightarrow$$

traditional interface can be considered outdated
(execution times are bounded for non-preempted execution)

[1] memory blocks a,b,c,d, * indicates cache miss, direct-mapped cache

Aim of this Thesis

A precise and sound

a) computation of the preemption cost, and

b) integration with schedulability analysis.

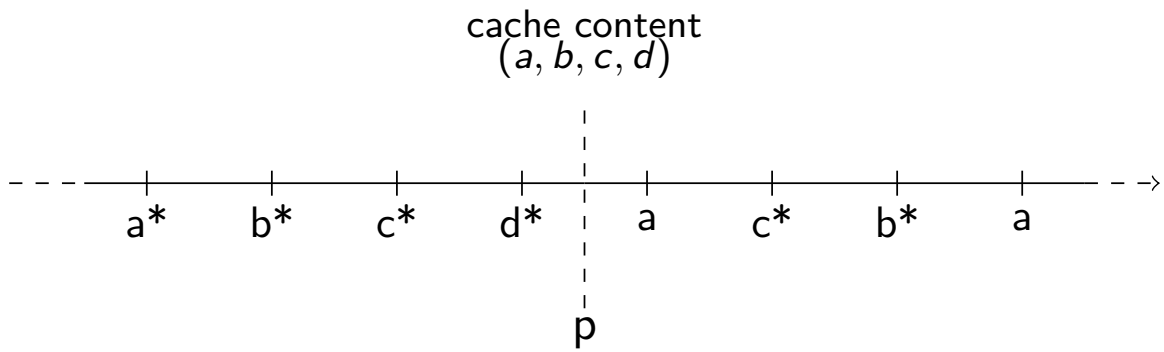
Useful Cache Blocks

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.

- $|UCB_P| \rightarrow$ CRPD bound at p
- $\max_P |UCB_P| \rightarrow$ global CRPD bound (based on preempted task)

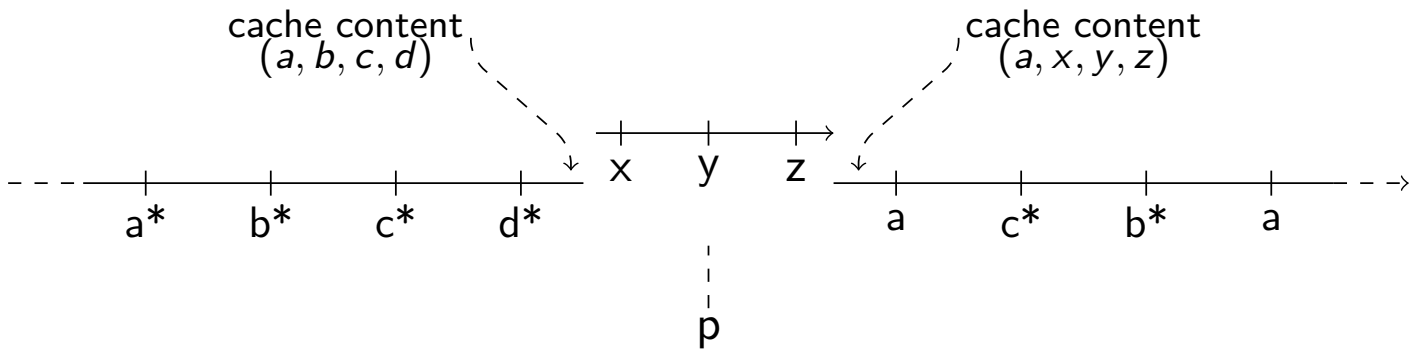


$$UCB_p = \{a, b, c\}$$

Evicting Cache Blocks

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.

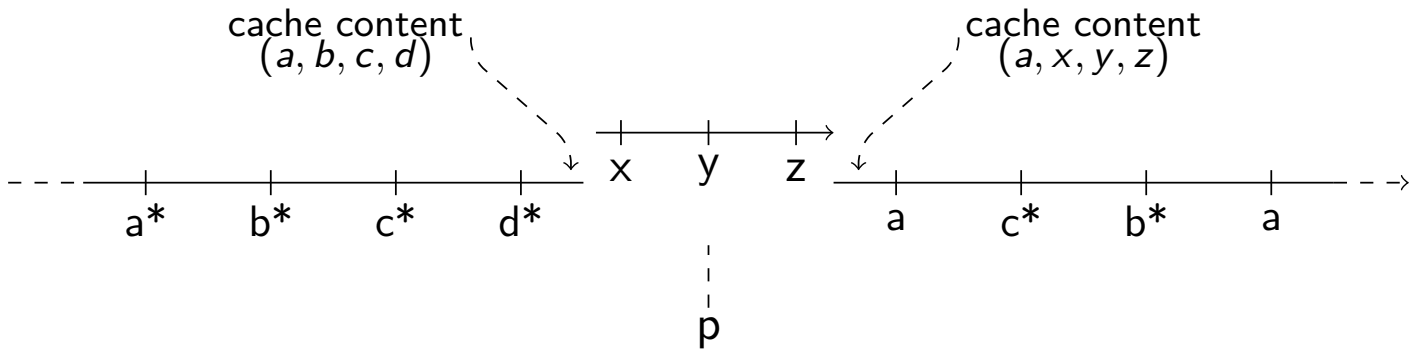
- $|ECB| \rightarrow$ global CRPD bound (based on preempting task)



$$ECB = \{x, y, z\}$$

Combining UCBs/ECBs

Only memory blocks that are both useful and will be evicted due to preemption lead to additional cache misses due to preemption.



$$UCB \sqcap ECB = \{b, c\}$$

Cache-Related Preemption Delay - Definition

Given an execution trace, a set of preemption points with preempting access sequence, initial cache state, and block reload time (BRT):

$$CRPD = (\#\text{misses on preempted execution} - \#\text{misses on non-preempted execution}) \cdot \text{BRT}$$

Difference to former definitions:

- based on concrete execution trace
- CRPD $\hat{=}$ total cost of all preemptions, not just for one preemption
- preemption cost defined with respect to the execution time

Cache-Related Preemption Delay - Limitations (1)

$$CRPD = (\#\text{misses on preempted execution} - \#\text{misses on non-preempted execution}) \cdot \text{BRT}$$

CRPD depends on block reload time BRT:

$\Rightarrow$

BRT must be bounded

$\Rightarrow$

Separate CRPD-computation not possible for
non-compositional architectures
e.g. architectures with timing anomalies.

Redefining Useful Cache Blocks

CRPD defined with respect to non-preempted execution time.
Why?

Schedulability analysis always computes:

$$WCET + n \times CRPD$$

n: # possible preemptions

$CRPD$: # $(UCB \sqcap ECB) \times$ BRT

However,
Timing analysis and UCB analysis are treated separately

Timing Analysis vs. UCB Analysis

Timing analysis:

- uses **underapproximation** of cache-content (**must-cache**)
- only predicts cache-hit, if accessed block $\in$ **must-cache**

UCB analysis:

- uses **overapproximation** of cache-content (**may-cache**)
- predicts additional cache-miss, if accessed block $\in$ **may-cache**

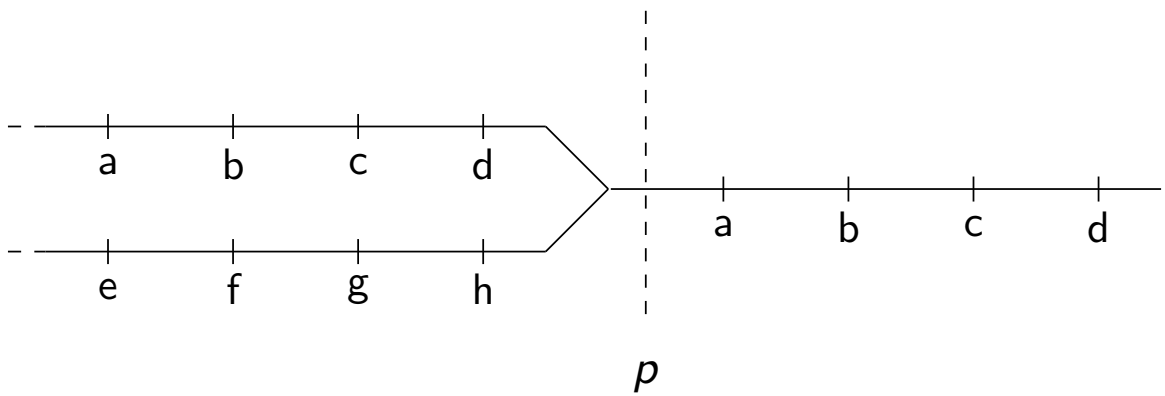
Hence, some cache-misses counted twice
(each access m with $m \in (\text{may-cache} \setminus \text{must-cache})$)

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, and

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



$$\mathrm{UCB}_p = \{a, b, c, d\}$$

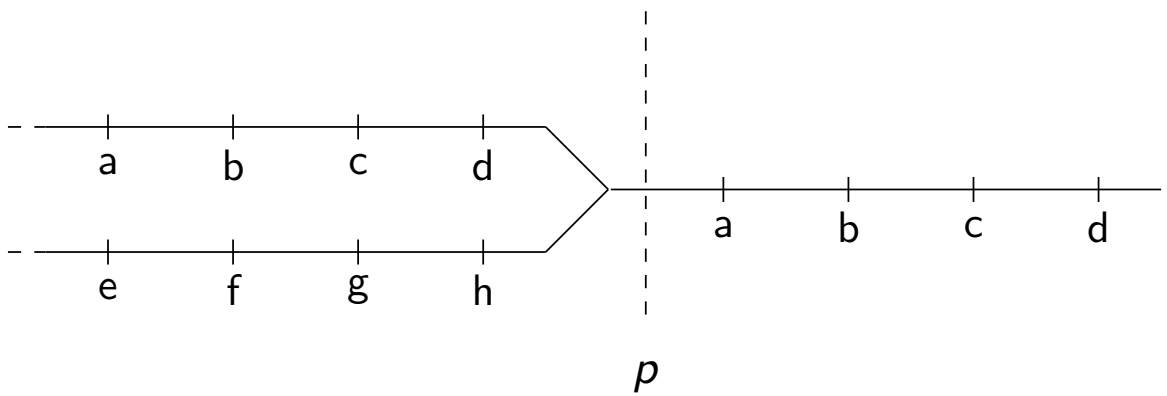
Definitely-Cached Useful Cache Blocks

A memory block m at program point p is called a definitely-cached useful cache block, if

a) m must 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, and

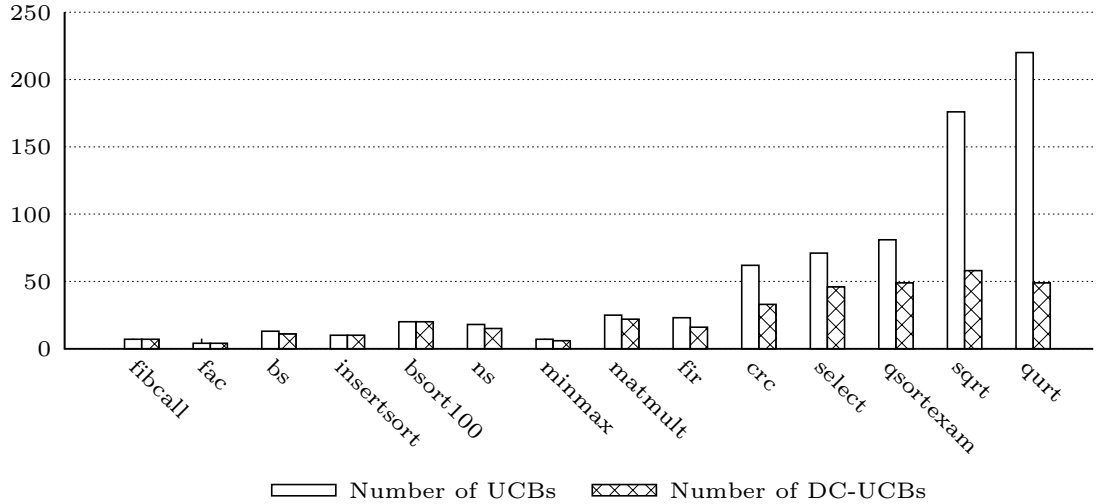
c) m is considered a cache hit at p' by the timing analysis.



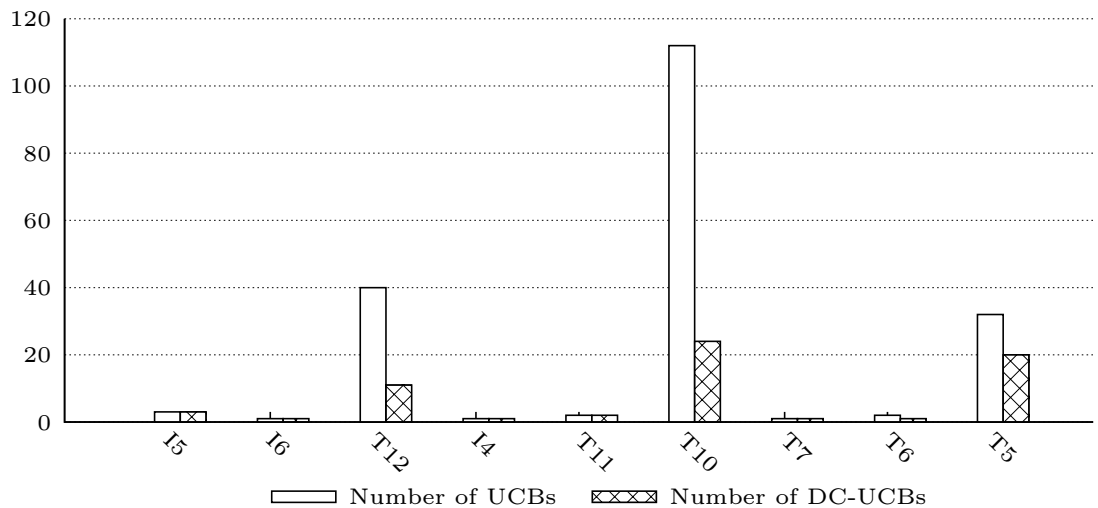
$$\text{DC-UCB}_p = \emptyset$$

Evaluation - DCUCBs

Mälardalen Benchmark Suite:



Papabench:



(ARM 7 with direct-mapped cache, 256 cache sets)

DC-UCBs and Timing Analysis

- DC-UCBs $\subseteq$ UCBs
- DC-UCBs do not bound CRPD in isolation
- but: DC-UCBs bound CRPD with respect to timing analysis

$\Rightarrow$

Concept of DC-UCBs strongly improve prior CRPD bounds

CRPD for Set-Associative Caches

What was already published?

CRPD bounds for

- LRU caches solely based on ECBs
- LRU/FIFO/PLRU caches solely based on UCBs
- LRU/FIFO/PLRU caches based on UCBs and ECBs

So everything solved?

Unfortunately, only LRU solely based on UCBs correct.

Cache-Related Preemption Delay - Limitations (2)

In case of FIFO and PLRU, CRPD (for a single preemption) is not bounded by the associativity of the cache but may be proportional to the total execution time.

$\Rightarrow$

Concept of UCBs/ECBs is not sufficient for FIFO/PLRU

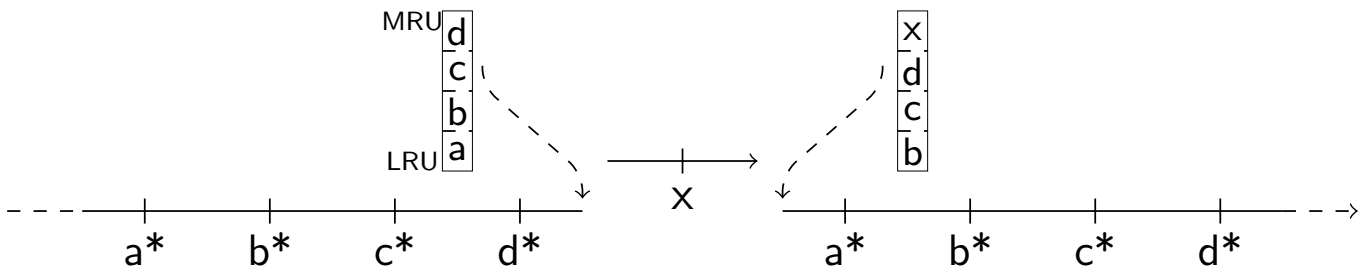
LRU Caches

In LRU caches, 2 different states converge after at most associativity-many distinct accesses

$\rightarrow$

associativity is always a valid upper bound.

Problem: one ECB suffices to remove all UCBs

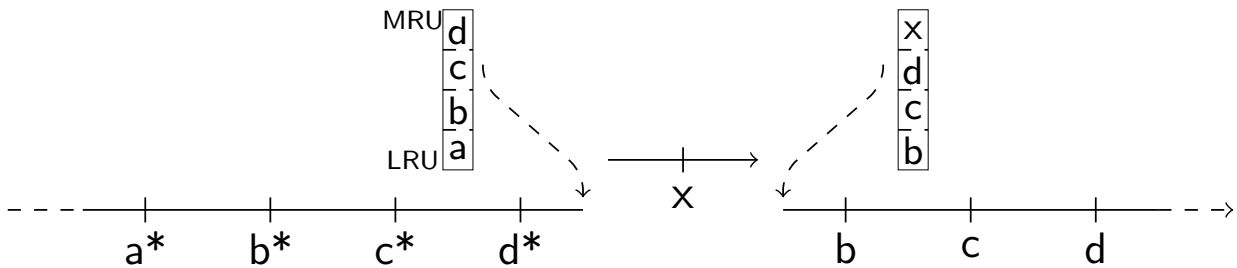


Only one ECB, but four additional misses
(former approaches fail here)

Simple CRPD Analysis for LRU

$$\text{CRPD} = \begin{cases} \text{BRT} \cdot |\text{UCB}(p)| & \text{if ECB} \neq \emptyset \\ 0 & \text{if ECB} = \emptyset \end{cases}$$

Sound but imprecise:



One ECB, three UCBs, but no additional misses.

In-depth Combination: Resilience Analysis

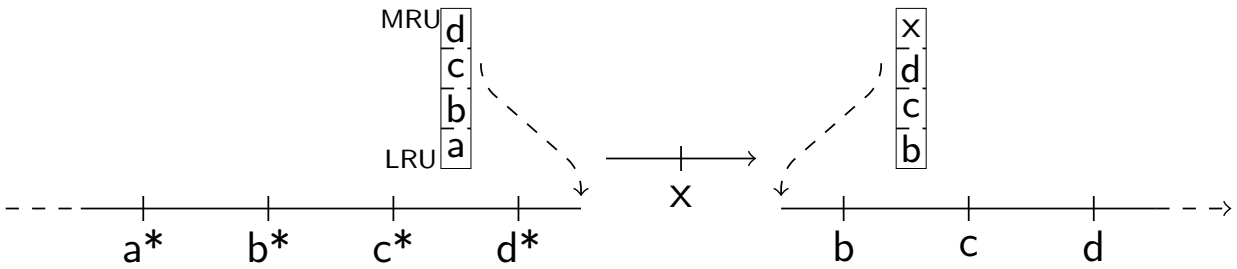
Resilience of UCB $m \mathrel{\hat{=}}$
amount of disturbance, s.t. m remains useful

Resilience of m at p is given by associativity k - age of m at next reuse p' of m.

$$\mathrm{res}_p(m) = \min(k - \mathrm{age}(m)_{p'})$$

$$\mathrm{CRPD} = \mathrm{BRT} \times |\overbrace{\underbrace{\mathrm{UCB}}_{\textit{useful}} \setminus \underbrace{\{m \mid \mathrm{res}(m) \geq |\mathrm{ECB}|\}}_{\textit{remain useful}}}^{\textit{blocks contributing to CRPD}}|$$

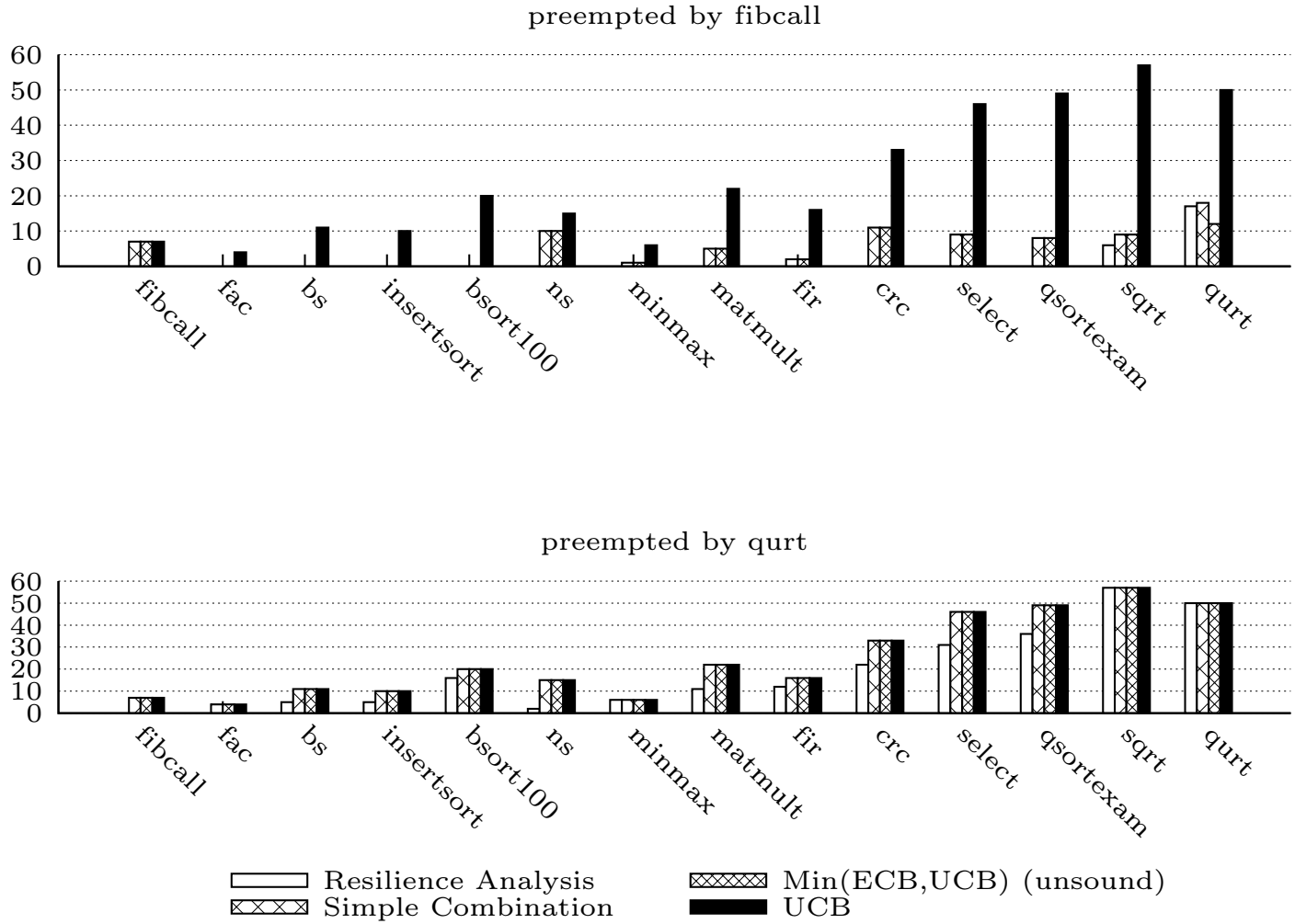
Resilience Analysis - Example



Blocks b,c,d are all 1-resilient,
i.e., survive preemption with up to one ECB

$$\begin{aligned}\text{CRPD} &= \text{BRT} \times |\text{UCB} \setminus \{m \mid \text{res}(m) \geq |\text{ECB}|\}| \\ &= \text{BRT} \times |\{b, c, d\} \setminus \{m \mid \text{res}(m) \geq 1\}| \\ &= \text{BRT} \times |\{b, c, d\} \setminus \{b, c, d\}| = 0\end{aligned}$$

Evaluation - Mälardalen Benchmark Suite



(ARM 7, 4-way LRU, 64 cache sets)

CRPD for Set-Associative Caches

All former approaches either imprecise or incorrect or both.

UCB/ECB approach not applicable to PLRU/FIFO Caches

Resilience Analysis provides

- provably correct CRPD bounds for LRU Caches
- precise combination of UCBs/ECBs

We now have precise CRPD bounds,
but how to use them?

In the following:

- Fixed priority scheduling
- Schedulability check via response time analysis (RTA)

Response Time Analysis

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

Response Time R_i = finishing time $-$ activation time

no deadline miss $\Leftrightarrow \forall i : R_i \leq D_i$

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

Response Time Analysis with CRPD

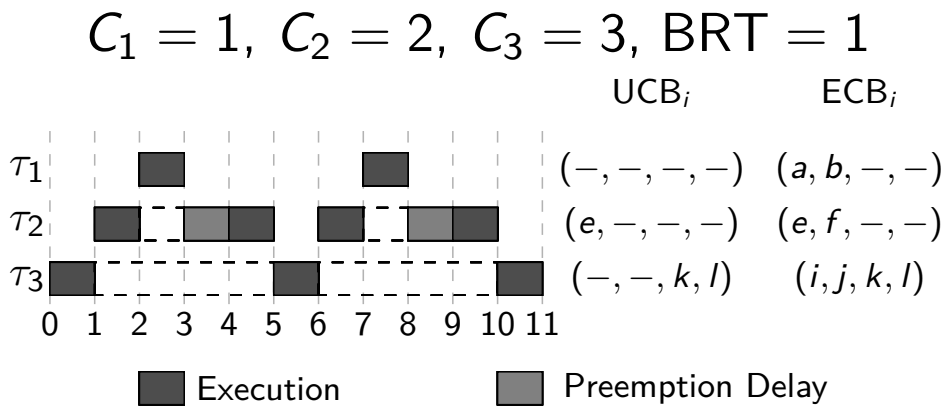
$$R_i = C_i + \sum_{\forall j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil (C_j + \gamma_{i,j})$$

$\gamma_{i,j}$ denotes the preemption cost

How to compute $\gamma_{i,j}$?

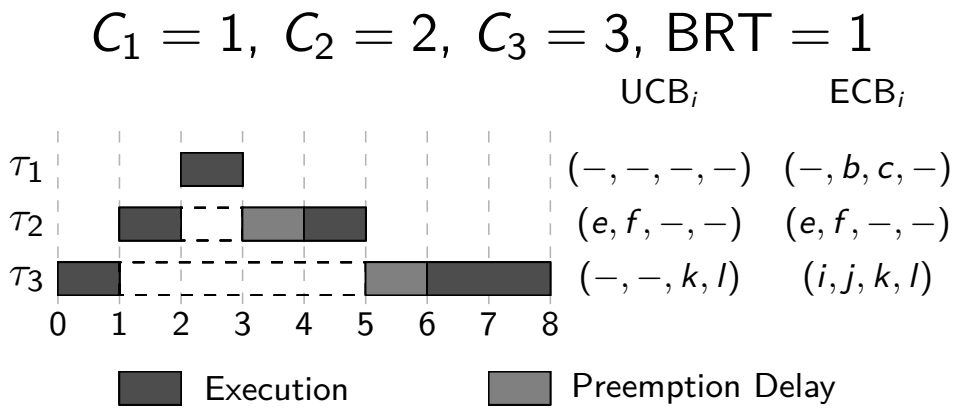
- Only counting UCBs of preempted task
- Only counting ECBs of preempting task
- Simple combination $|\mathrm{UCB}_i \sqcap \mathrm{ECB}_j|$ possible?

Pitfall 1: Preemption of Intermediate Tasks



When computing R_3:
τ_1 preempting τ_2 causes higher costs (1) than τ_1 preempting τ_3 (0)

Pitfall 2: Nested Preemption



When computing R_3:
Nested preemption causes higher costs (2) than any non-nested (1)

New Approach: ECB-Union

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

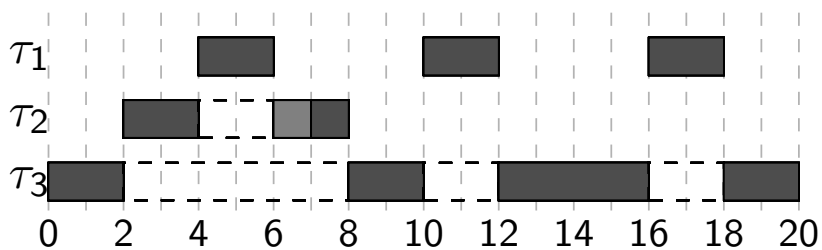
- safe combination of ECBs and UCBs
- dominates UCB-Only
- resilience analysis fits well into ECB-Union approach (resilience needs ECB-Union of all prior preemptions)

Pessimism in basic RTA Formula

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

$\gamma_{i,j}$ = preemption cost of one job of τ_j during R_i

---



	UCB_i	ECB_i
τ_1	$(a, b, -, -)$	$(a, b, -, -)$
τ_2	$(e, f, -, -)$	$(e, f, -, -)$
τ_3	$(-, -, k, l)$	(i, j, k, l)

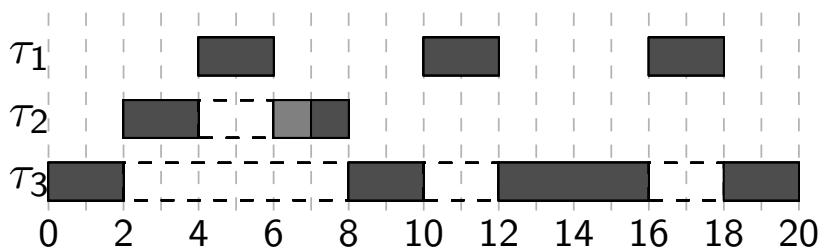
τ_1 preempting τ_2 considered 3 times within R_3

Pessimism in basic RTA Formula

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

$\gamma_{i,j}$ = preemption cost of all jobs of τ_j during R_i

---



	UCB_i	ECB_i
τ_1	$(a, b, -, -)$	$(a, b, -, -)$
τ_2	$(e, f, -, -)$	$(e, f, -, -)$
τ_3	$(-, -, k, l)$	(i, j, k, l)

τ_1 preempting τ_2 considered only once within R_3

Multiset Approaches

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

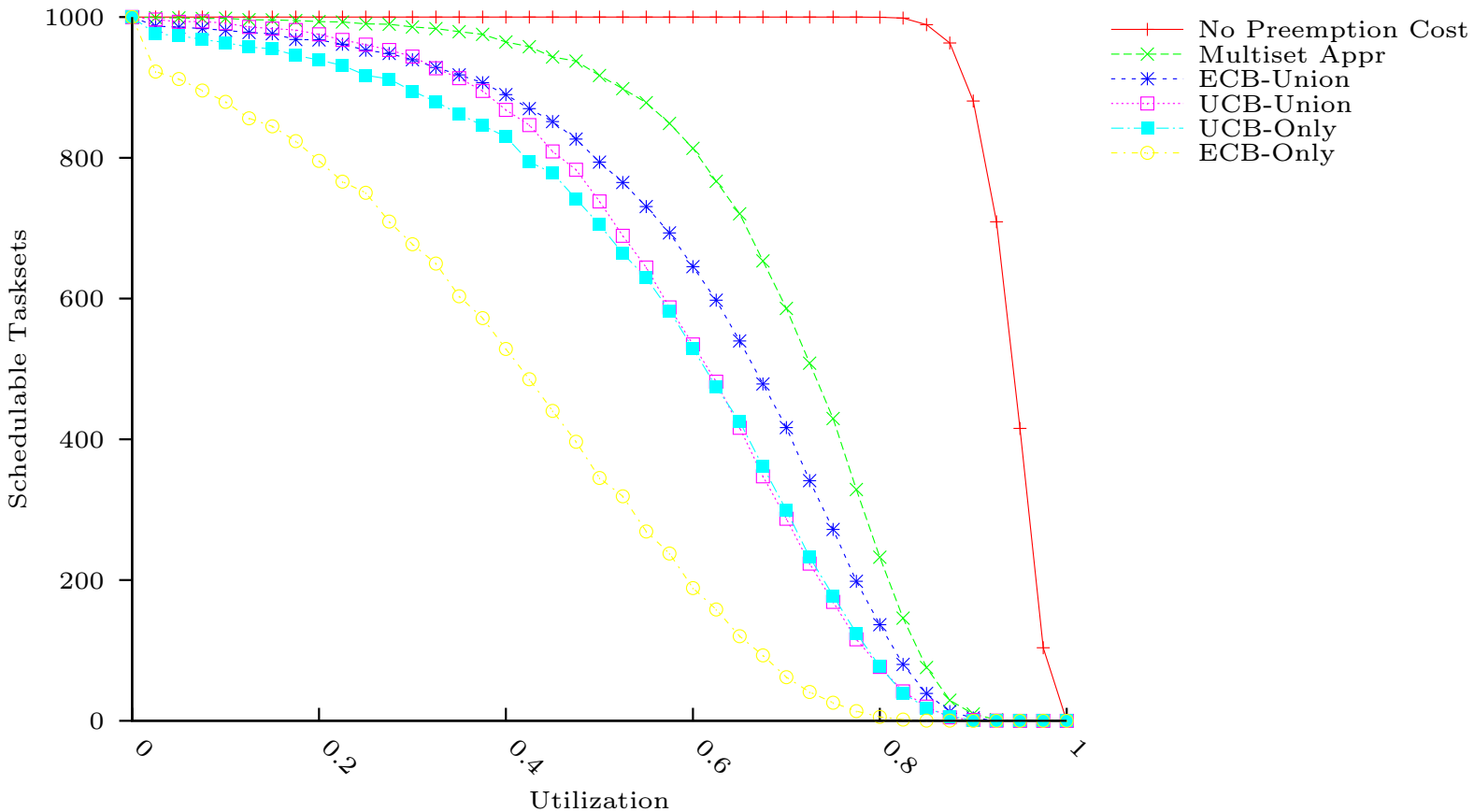
$\gamma_{i,j}$ = preemption cost of all jobs of τ_j during R_i

---

$$M = \bigcup_{k \in \mathsf{aff}(i,j)} \left(\underbrace{\bigcup_{\lceil R_k / T_j \rceil \lceil R_i / T_k \rceil}}_{\#\ \tau_j \text{ preempts } \tau_k \text{ during } R_i} \mathrm{Cost}_{k,j} \right) \qquad \gamma_{i,j} = \mathrm{BRT} \cdot \underbrace{\sum_{l=1}^{\lceil R_i / T_j \rceil} M^l}_{\text{highest values in } M}$$

M is a multiset, $\mathrm{Cost}_{k,j}$ can be computed by any correct former approach

Evaluation



Randomly generated task sets with 10 tasks, implicit deadline, 256 cache sets, $CRT = 8\mu s$, cache utilization of 10

Conclusions

Aim: A precise and sound

a) computation of the preemption cost, and

b) integration with schedulability analysis.

Conclusions

Aim: A precise and sound

a) computation of the preemption cost, and

b) integration with schedulability analysis.

Results:

a) Preemption cost computation:
   - formal definition of CRPD
   - description of limitations of the CRPD approach
   - concept of Definitely-Cache Useful Cache Blocks
   - CRPD for LRU Caches (Resilience Analysis)

Conclusions

Aim: A precise and sound

a) computation of the preemption cost, and

b) integration with schedulability analysis.

Results:

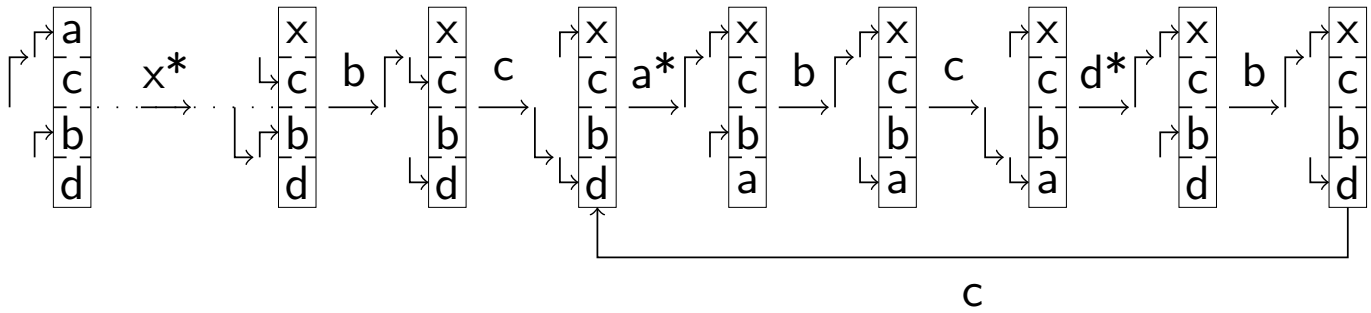
a) Preemption cost computation:
   - formal definition of CRPD
   - description of limitations of the CRPD approach
   - concept of Definitely-Cache Useful Cache Blocks
   - CRPD for LRU Caches (Resilience Analysis)

b) CRPD-aware schedulability analysis:
   - ECB-Union approach
   - Multiset extensions
   - correct consideration of the blocking time

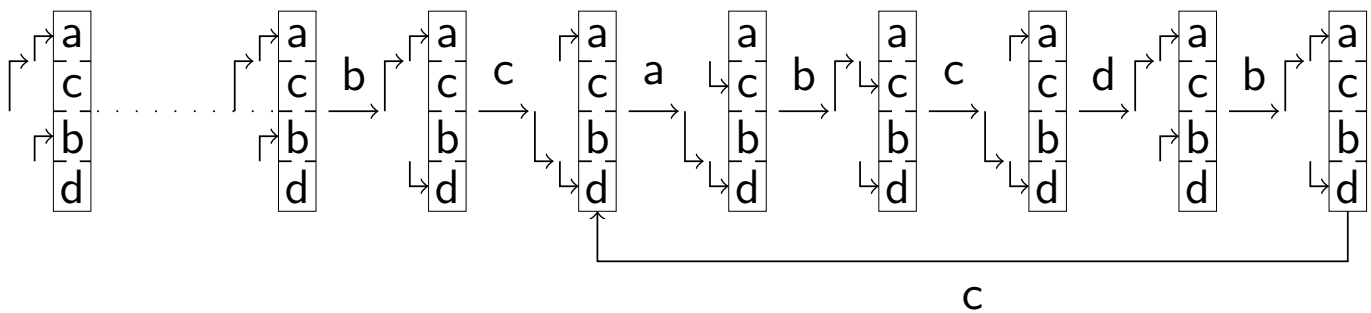
Questions?

CRPD - Limitations (2) - PLRU

Preempted execution ($2 * n$ misses):

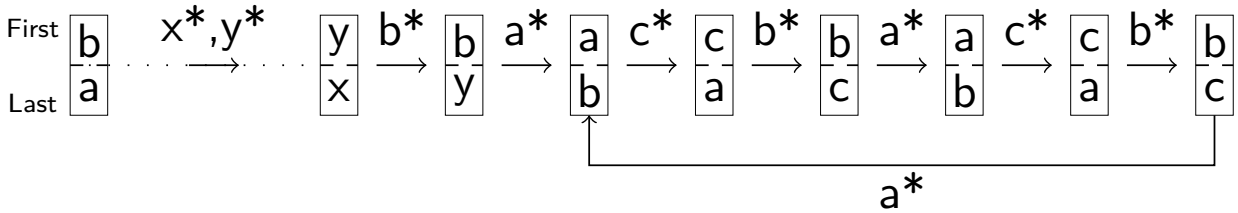


Non-preempted execution (0 misses):

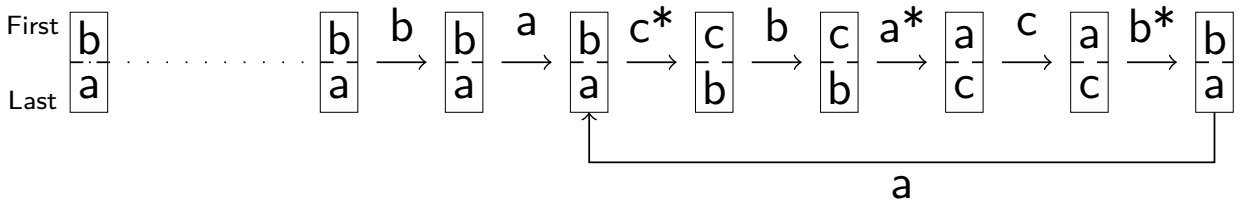


CRPD - Limitations (2) - FIFO

Preempted execution ($2 + 5 * n$ misses):



Non-preempted execution ($3 * n$ misses):



DC-UCB Analysis - Property on Concrete Paths

For Each memory block $m \in \text{DC-UCB}(p)$, exists a path s.t. m is reused at its next access p_n and m is classified as always hit at p_n:

$$m \in \text{DC-UCB}(p_i) \Leftrightarrow \exists [p_i, \ldots, p_n] \in \Pi : \\ \sharp(p_n) = m \wedge \text{Classify}(m, p_n) = ah \wedge \forall_{j=i}^{n-1} \sharp(p_j) \neq m \quad (1)$$

$$\text{Els} : V \times M \rightarrow \{\text{true}, \text{false}\}$$

$$\text{Els}(\pi, m) = \begin{cases} \text{true} & \text{if } \pi = [p_1, \ldots, p_n] \cdot \pi' \wedge \sharp(p_n) = m \\ & \quad \wedge \text{Classify}(m, p_n) = ah \wedge \forall_{j=i}^{n-1} \sharp(p_j) \neq m \\ \text{false} & \text{otherwise} \end{cases} \quad (2)$$

DC-UCB - Abstract Transformer

$$tf : V \to (2^M \to 2^M)$$

$$tf(p)(S) = \begin{cases} S & \sharp(p) = \bot \\ S \cup \{\sharp(p)\} & \text{Classify}(\sharp(p), p) = ah \\ S \setminus \{\sharp(p)\} & \text{Classify}(\sharp(p), p) \neq ah \end{cases} \quad (3)$$

DC-UCB Analysis - Concretization/Abstraction

$$\gamma : 2^M \to 2^\Pi$$

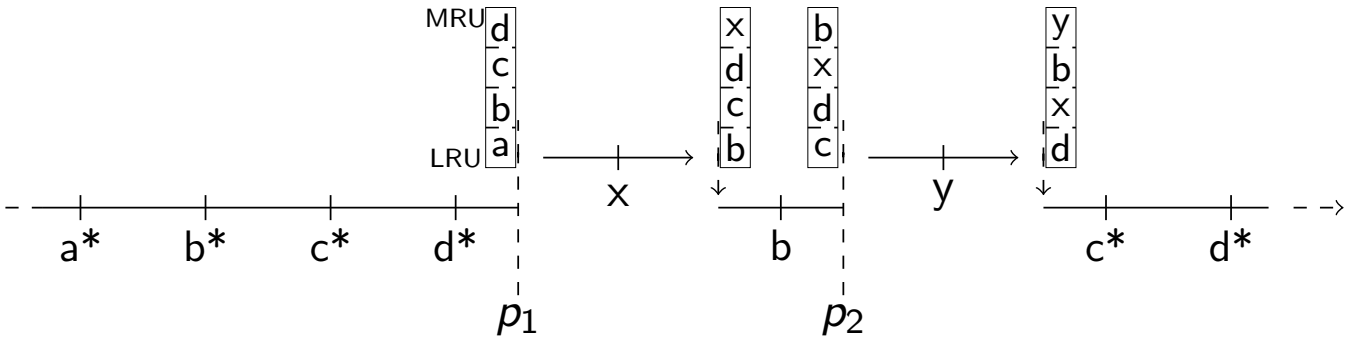
$$\gamma(S) := \{\pi \mid \pi \in \Pi : \forall m \in (M \setminus S) : \neg \mathsf{Els}(\pi, m)\} \tag{4}$$

$$\alpha : 2^\Pi \to 2^M$$

$$\alpha(S) := \{m \mid \exists \pi \in S : \mathsf{Els}(\pi, m)\} \tag{5}$$

Resilience Analysis - Interacting Preemptions

Problem: preemptions may interact



$\text{CRPD}_{p_1} = \text{CRPD}_{p_2} = 0$, but total CRPD $= 2$

Solution: Compute ECB-Union of all prior preemptions PR

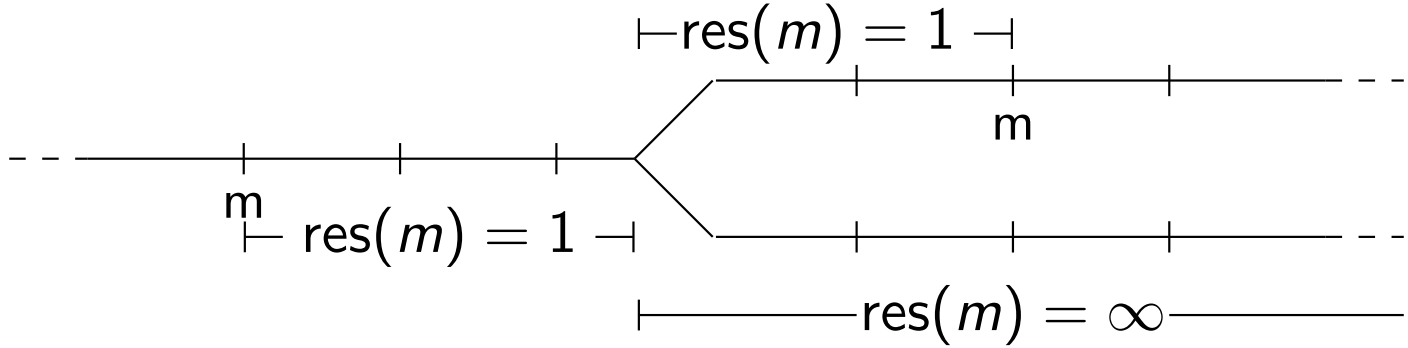
$$\text{CRPD} = \text{BRT} \times |\text{UCB} \setminus \{m \mid \text{res}(m) \geq |\bigcup_{PR} \text{ECB}|\}|$$

Example:

$$\text{CRPD}_{p_1} = \text{BRT} \times |\text{UCB} \setminus \{m \mid \text{res}(m) \geq |\{x\}|\}| = 0$$

$$\text{CRPD}_{p_2} = \text{BRT} \times |\text{UCB} \setminus \{m \mid \text{res}(m) \geq |\{x, y\}|\}| = 2$$

Resilience under Different Paths



Resilience under different paths, assuming a 4 way LRU cache.

Resilience Analysis - Constrained/Unconstrained Age

$$tf_{ua} : V \to \text{Age} \to \text{Age}$$

$$tf_{ua}(p)(ua) := \lambda m. \begin{cases} 0 & m = \sharp(p) \\ ua(m) & ua(m) \geq ua(\sharp(p)) \\ ua(m)+1 & ua(m) < ua(\sharp(p)) \wedge ua(m) < ua - 1 \\ \infty & otherwise \end{cases} \tag{6}$$

The accessed element $\sharp(p)$ is assigned age zero, all younger elements (as the accessed element) age by one.

$$tf_{ca} : V \to (\text{Age} \times \text{Age} \to \text{Age})$$

$$tf_{ca}(p)(ca, ua) := \lambda m. \begin{cases} 0 & m = \sharp(p) \wedge m \notin \text{UCB}(p) \\ ca(m) & ca(m) \geq ua(\sharp(p)) \vee ca(m') = k - 1 \\ ca(m)+1 & ca(m) < ua(\sharp(p)) \end{cases} \tag{7}$$

Resilience - Concretization/Abstraction

$$\gamma^{\rightarrow}((ua, ca)) = \{(p \cdot \pi) \in \Pi \mid \forall m \in M : (\widehat{\text{age}}^{\rightarrow}(p \cdot \pi)(m) \leq ca^{\rightarrow}(m)) \\ \vee (\widehat{\text{age}}^{\rightarrow}(p \cdot \pi)(m) \leq ua^{\rightarrow}(m) \wedge m \notin \text{UCB}_p(m))\} \quad (8)$$

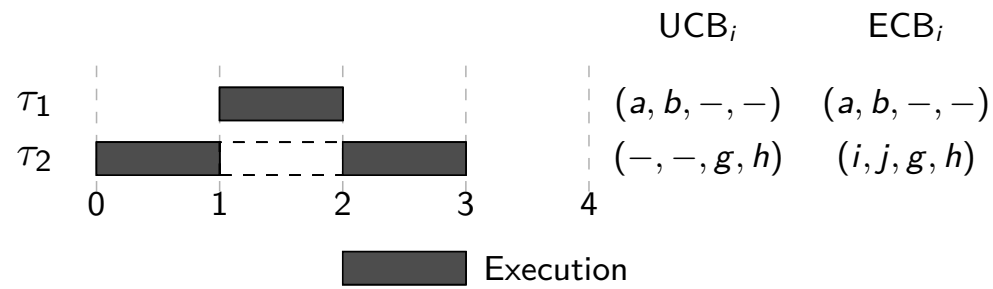
$$\alpha^{\rightarrow}(S) = (\lambda m. \max\{\widehat{\text{age}}^{\rightarrow}(p \cdot \pi)(m) \mid (p \cdot \pi) \in S\}, \\ \lambda m. \max\{\widehat{\text{age}}^{\rightarrow}(p \cdot \pi)(m) \mid (p \cdot \pi) \in S \wedge m \in \text{UCB}(p)\}) \cup \{0\}) \quad (9)$$

Prior Work: UCB only or ECB only

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

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

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



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

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

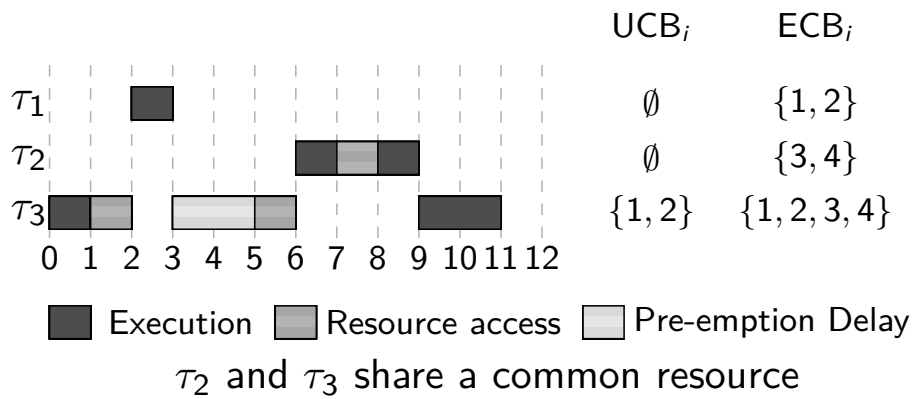
Prior Work: UCB-Union

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

- safe combination of ECBs and UCBs
- dominates ECB-Only ($\gamma_{i,j}^{\text{ecb}} = \text{BRT} \cdot |\text{ECB}_j|$)
- poor precision: set UCBs per task, not per program point

Blocking Time (Stack Resource Protocol)

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



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

Evaluation - Cache Configurations

	Cache Type	Sets	Line Size	Total Size
Config 1:	direct-mapped	256	16 Byte	4kB
Config 2:	4-way LRU	64	16 Byte	4kB
Config 3:	8-way LRU	64	16 Byte	8kB

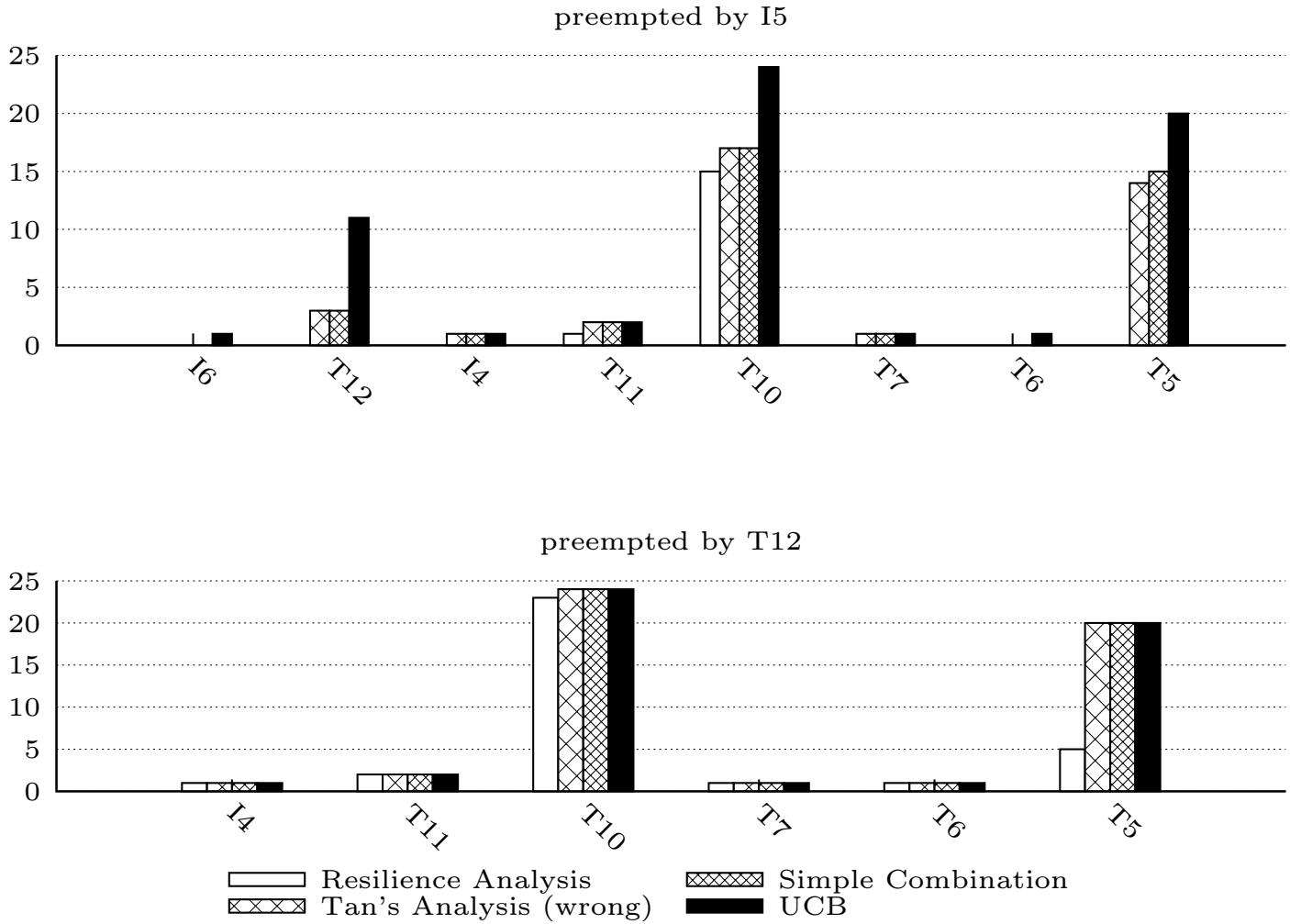
Mälardalen Benchmark Suite

Task	Description	Code Size	WCET
minmax	Derives min/max of set of integers	$608B$	$298\mu s$
insertsort	Insertion sort on array of size 10.	$384B$	$223\mu s$
fibcall	Iterative Fibonacci calculation).	$256B$	$117\mu s$
fac	Non-recursive Faculty Computation	$256B$	$67\mu s$
bs	Binary search (array of size 15).	$320B$	$1.46ms$
bsort100	Bubblesort program.	$544B$	$11.3ms$
ns	Test for deeply nested loops	$576B$	$637\mu s$
matmult	Matrix multiplication (20x20).	$864B$	$8.5ms$
fir	Finite impulse response filter.	$928B$	$465\mu s$
crc	Cyclic redundancy check .	$1216B$	$2.7ms$
select	Selects Nth largest number.	$1280B$	$757\mu s$
qsort-exam	Non-recursive quick sort algorithm.	$1440B$	$800\mu s$
sqrt	Square root by Taylor series.	$3680B$	$2.1ms$
qurt	Computation of quadr. equations.	$4160B$	$2.9ms$

Papabench

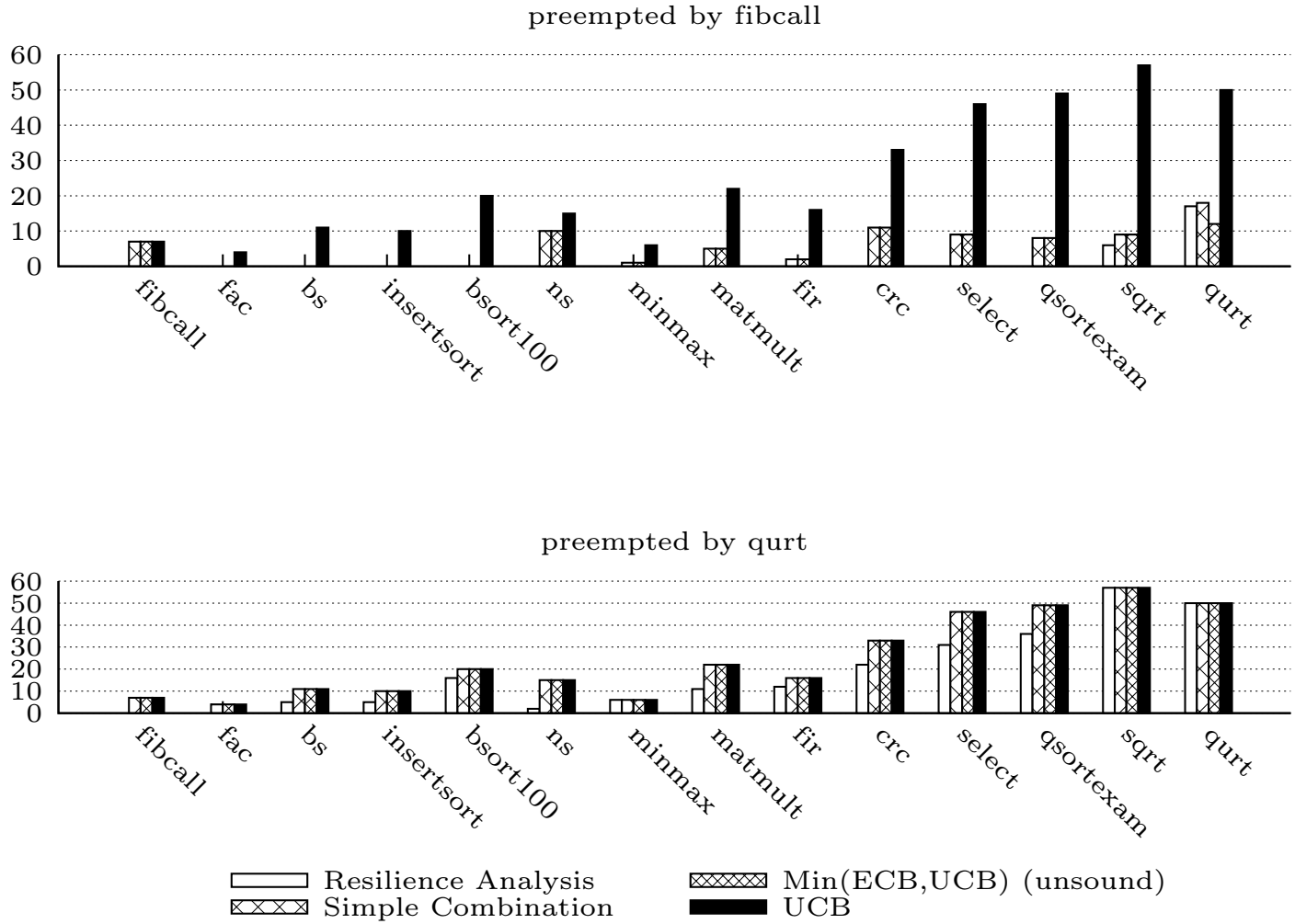
Task	Description	Period	Priority	Code Size	WCET
I5	interrupt_spi_1	$50ms$	1	$304B$	$129\mu s$
I6	interrupt_spi_2	$50ms$	2	$144B$	$68\mu s$
T12	stabilization	$50ms$	3	$2976B$	$3.2ms$
I4	interrupt_modem	$100ms$	4	$288B$	$148\mu s$
T11	reporting_task	$100ms$	5	$5360B$	$5.9ms$
T10	receive_gps_data	$250ms$	6	$3008B$	$3ms$
T7	link_fbw_send	$250ms$	7	$192B$	$105\mu s$
T6	climb_control	$250ms$	8	$3296B$	$3.4ms$
T5	altitude_control	$250ms$	9	$1232B$	$826\mu s$

Evaluation - Papabench



(ARM 7, 4-way LRU, 64 cache sets)

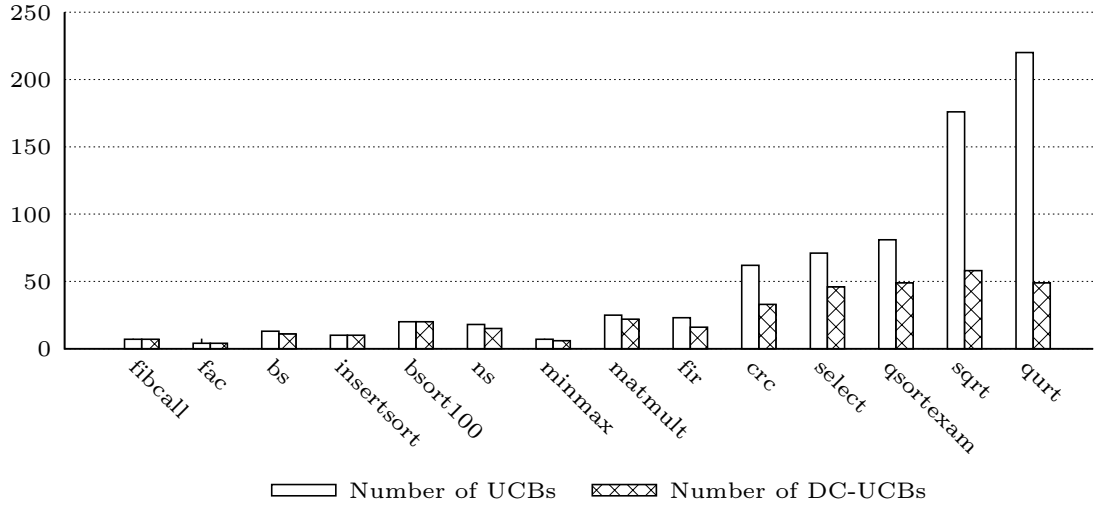
Evaluation - Mälardalen Benchmark Suite



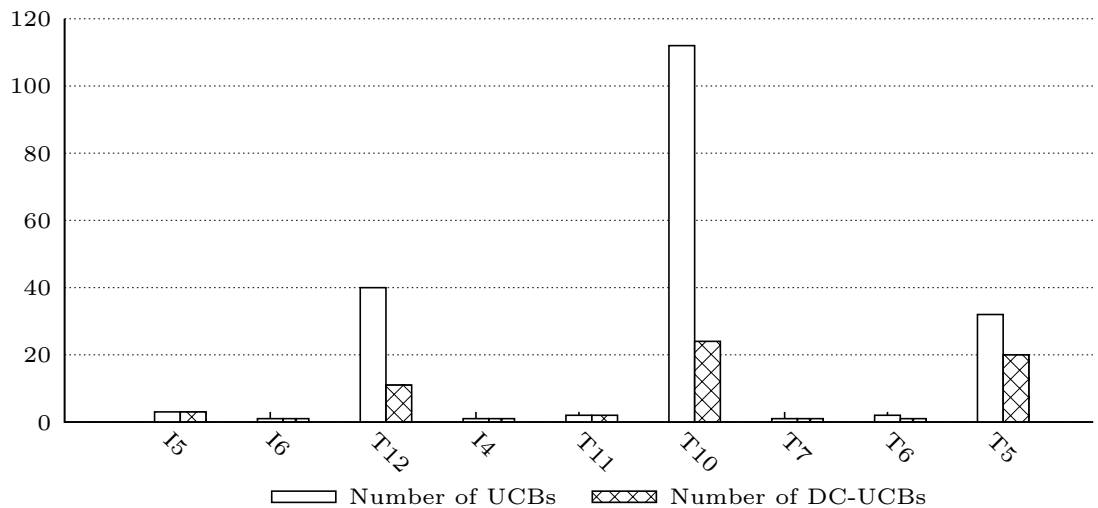
(ARM 7, 4-way LRU, 64 cache sets)

Evaluation - DCUCBs

Mälardalen Benchmark Suite:



Papabench:



(ARM 7 with direct-mapped cache, 256 cache sets)

Evaluation - Setting

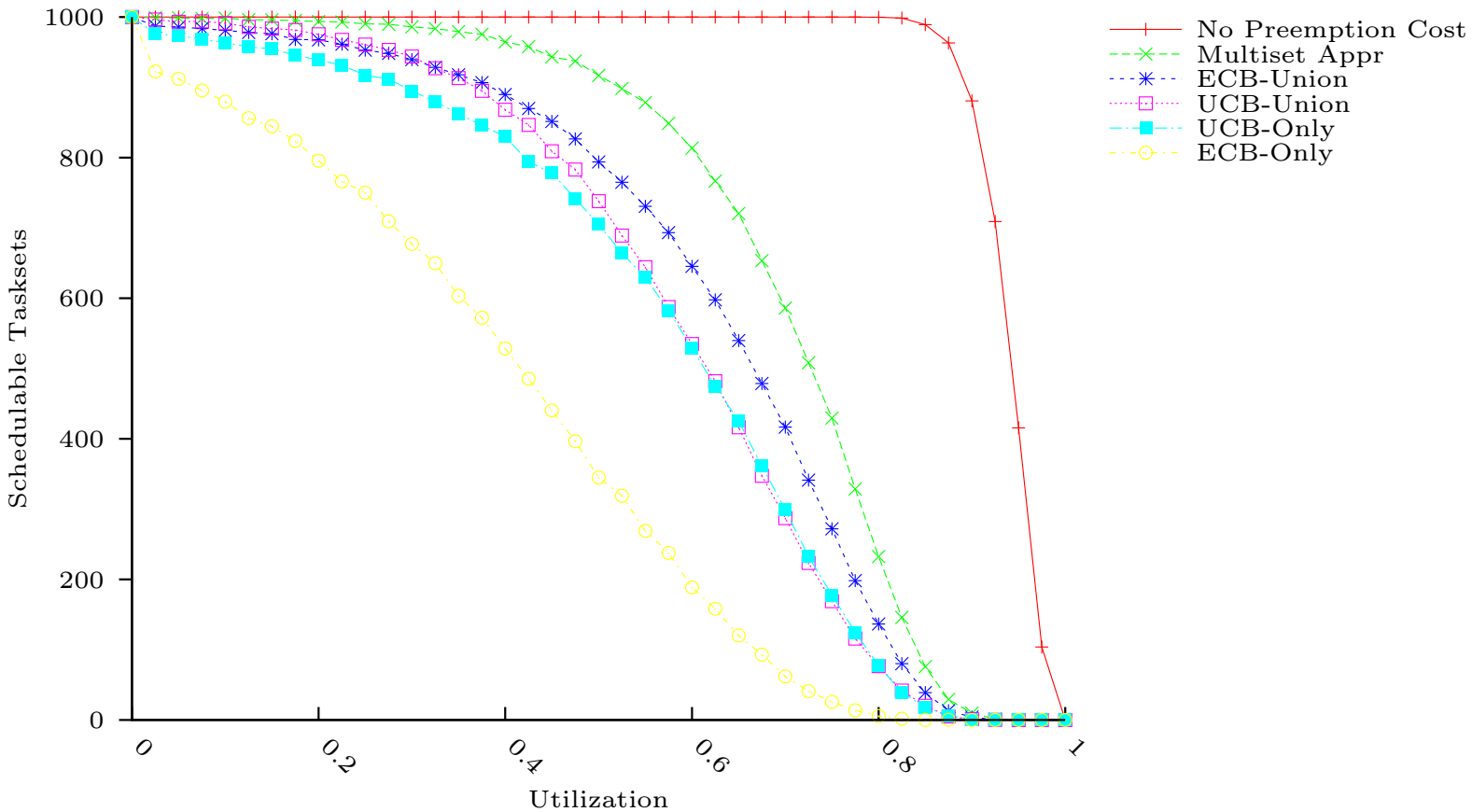
Task set:

- 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

Preemption costs:

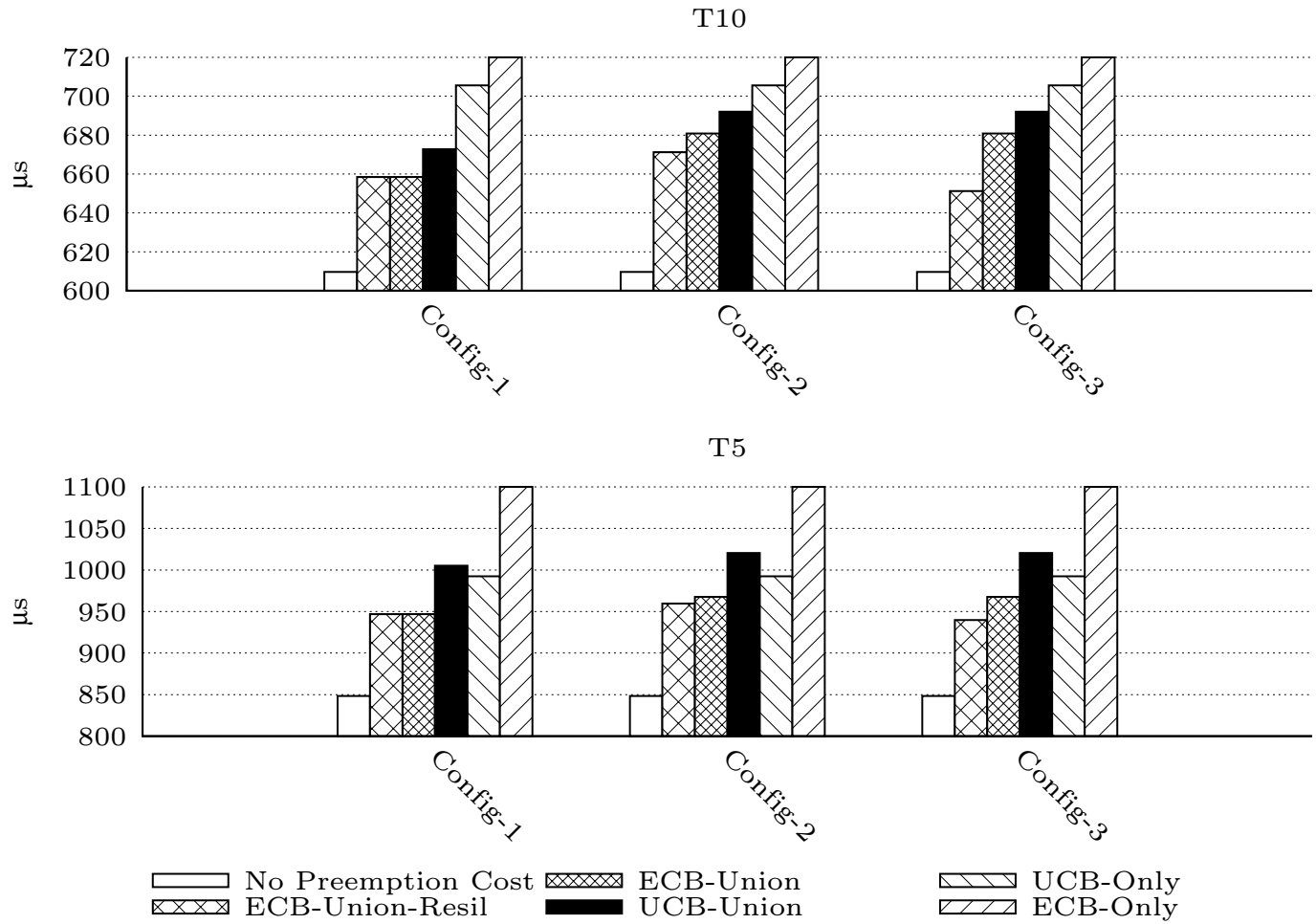
- number of cache sets ($CS = 256$)
- block-reload time ($CRT = 8\mu s$)
- cache usage using UUnifast ($CU = 10$)
- reuse factor (UCBs), uniform distribution $[0; |\text{ECB}|]$

Evaluation



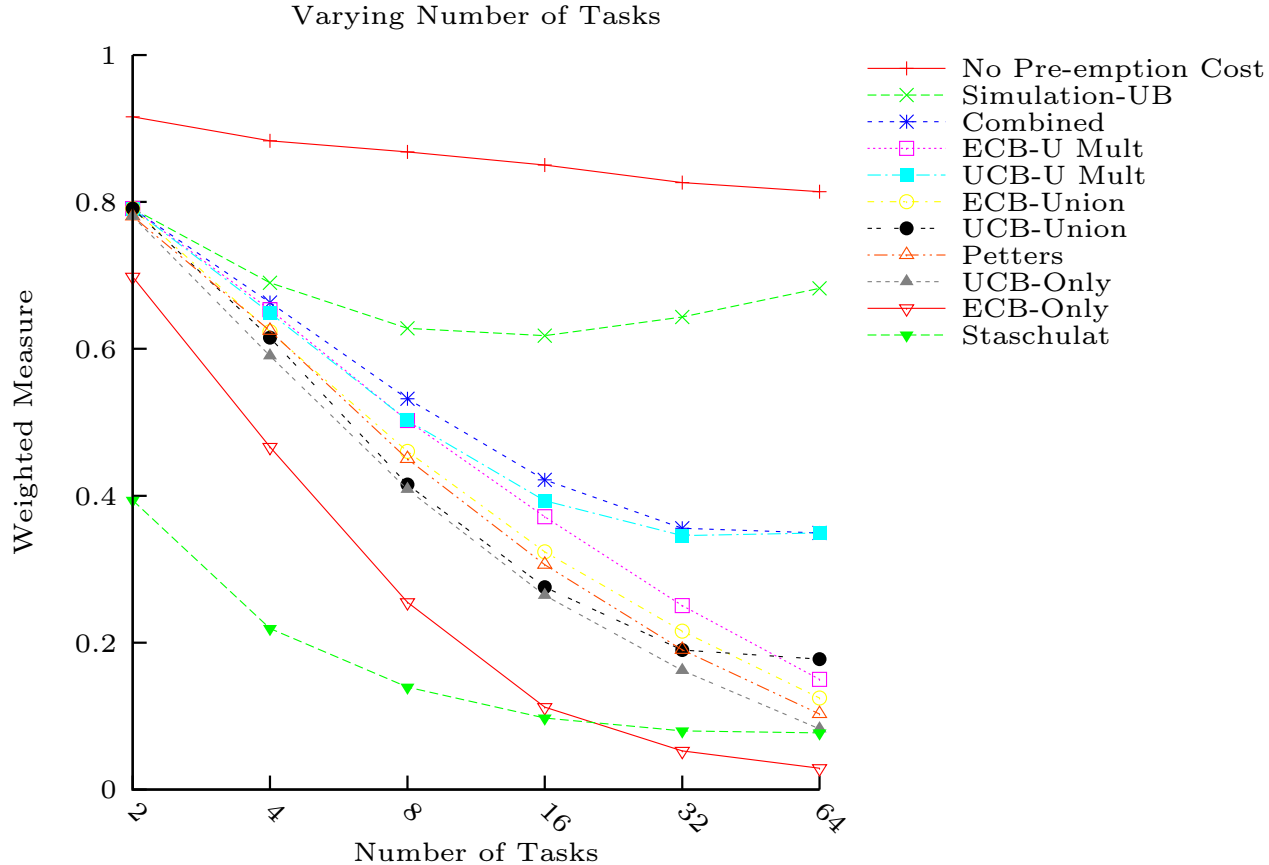
Randomly generated task sets with 10 tasks, implicit deadline, 256 cache sets, $CRT = 8\mu s$, cache utilization of 10

Evaluation - CRPD-Aware RTA



Bounds on the response time of T5 and T10 (in μs)

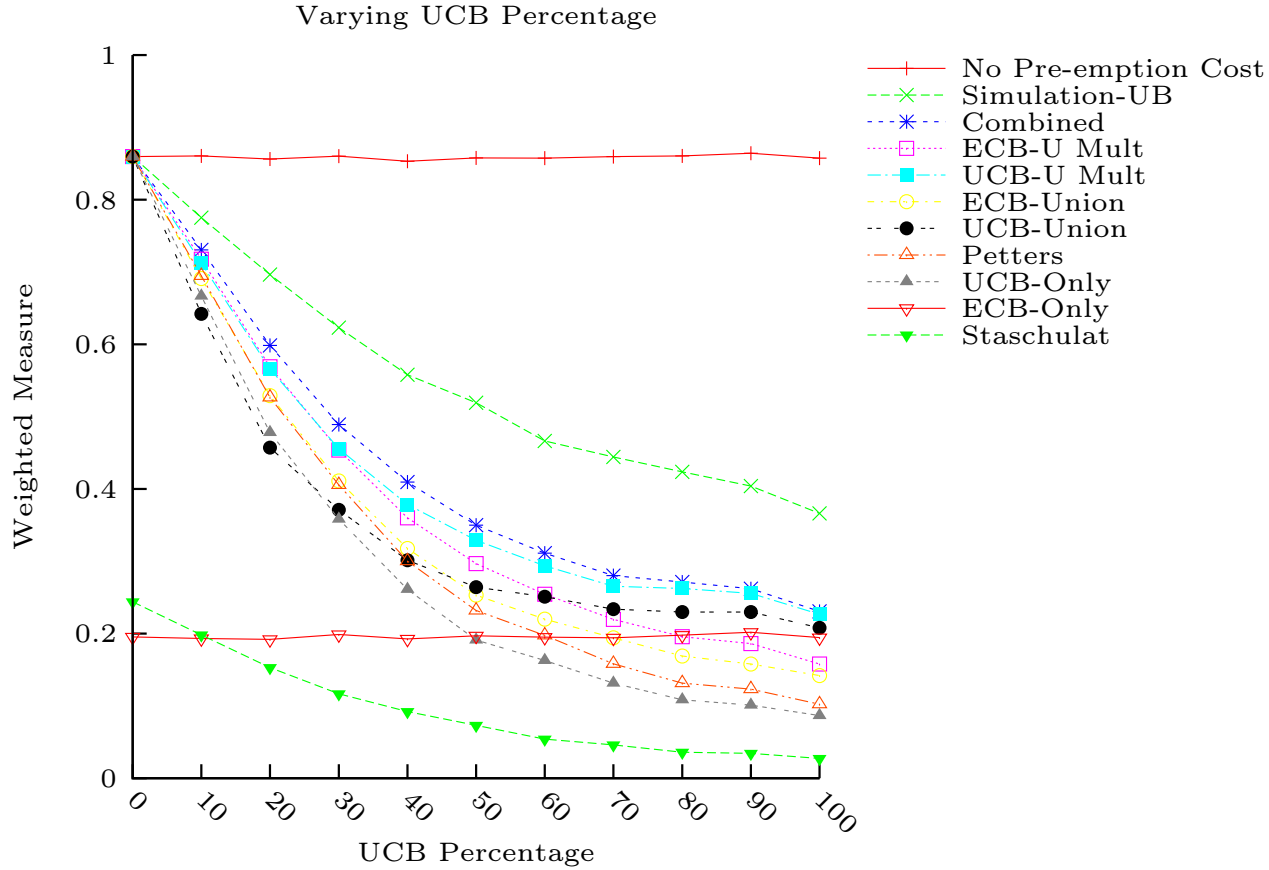
Evaluation - CRPD-Aware RTA - Number of Tasks



Weighted schedulability measure; varying number of tasks from $2 = 2^1$ to $2^6 = 64$.

$$W_y(p) = (\sum_{\forall \tau} u(\tau) \cdot S_y(\tau, p)) / \sum_{\forall \tau} u(\tau) \qquad (10)$$

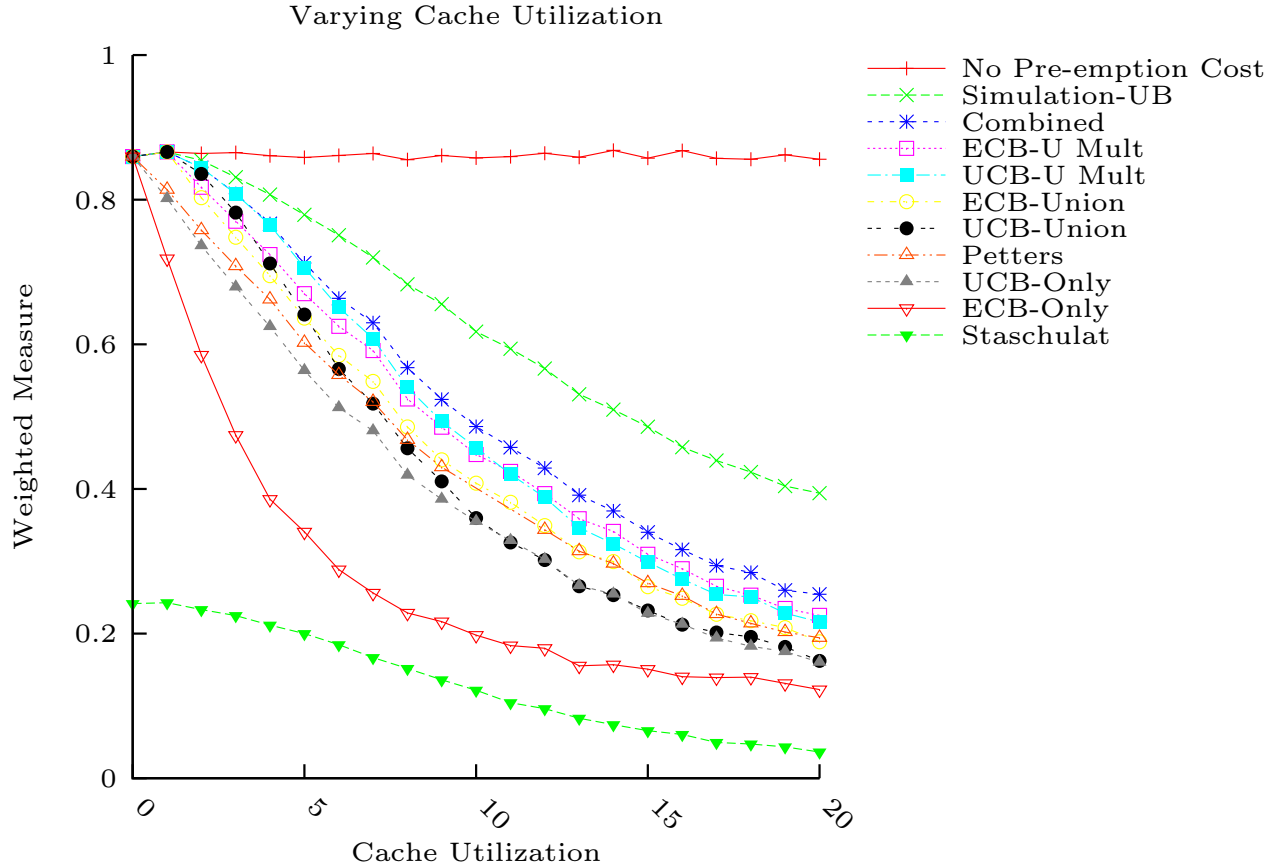
Evaluation - CRPD-Aware RTA - Reuse Factor



Weighted schedulability measure; varying reuse factor from 0% to 100%, in steps of 10%.

$$W_y(p) = (\sum_{\forall \tau} u(\tau) \cdot S_y(\tau, p)) / \sum_{\forall \tau} u(\tau) \qquad (11)$$

Evaluation - CRPD-Aware RTA - Cache Utilization



Weighted schedulability measure; varying cache utilization from 0 to 20, in steps of 2.

$$W_y(p) = \left(\sum_{\forall \tau} u(\tau) \cdot S_y(\tau, p)\right) / \sum_{\forall \tau} u(\tau) \qquad (12)$$