PROARTIS

Multi-level Unified Caches for Probabilistically Analysable Real-time Systems

Leonidas Kosmidis, Jaume Abella,
Eduardo Quiñones, Francisco J. Cazorla

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

This project and the research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 249100.

Outline

- Motivation
- Background
  - Probabilistic Timing Analysis
    - Measurement-Basted Timing Analysis requirements on HW
  - Time-Randomised Caches
  - Cache concepts: inclusivity, write-miss and allocation policies
- MBPTA for multilevel caches
  - NIC, WT-nWA caches
- Generalisation of the model
- Results
- Conclusions

Outline

- Motivation
- Background
  - Probabilistic Timing Analysis
    - Measurement-Basted Timing Analysis requirements on HW
  - Time-Randomised Caches
  - Cache concepts: inclusivity, write-miss and allocation policies
- MBPTA for multilevel caches
  - NIC, WT-nWA caches
- Generalisation of the model
- Results
- Conclusions

Caches in high-perf. and real-time systems

- *Requirements on caches in high-performance systems*
  - Have shown to improve average performance
  - Several level of caches deploy in real products
- *Requirements on caches in Real-time systems*
  - Also improve average performance
  - We aim at improving (i.e. reducing) guaranteed performance.
    - Reduces WCET estimates for tasks
    - Increases the load that the system accepts
    - Enables putting more functions on less hardware → Reduction of SWaP costs

Caches in real-time systems: The challenge

- *Challenge:*
  - Determining WCET estimates for tasks in a processor deploying several level of caches
- *Risk:*
  - If WCET estimates are not tight we defy the whole point of using L2 caches to increase guaranteed performance
- *Static Timing Analysis Approach:*
  - Cache analysis deemed tough for single-level caches
    - Mainly for data caches (predicting runtime data accesses is hard)
  - Few works attack the multilevel cache challenge[1][2]
    - Work for separated data and instruction L2 cache

[1] D. Hardy et al, WCET Analysis of multi-level non-inclusive instruction caches, RTSS '08

[2] B. Lesage et al, WCET Analysis of multi-level set-associative data caches, WCET '09

Probabilistic Timing Analysis and caches

- *PTA requires time randomised caches*
  - Cache for which each access has a a hit/miss probability
- *Time randomised caches have properties that enable the analysis of several levels of cache*

We show how MBPTA can analyse multilevel unified caches

Variable number of levels
Different inclusion policies
Different write-miss policies
Different write-allocation policies

Outline

- Motivation
- Background
  - Probabilistic Timing Analysis
    - Measurement-Basted Timing Analysis requirements on HW
  - Time-Randomised Caches
  - Cache concepts: inclusivity, write-miss and allocation policies
- MBPTA for multilevel caches
  - NIC, WT-nWA caches
- Generalisation of the model
- Results
- Conclusions

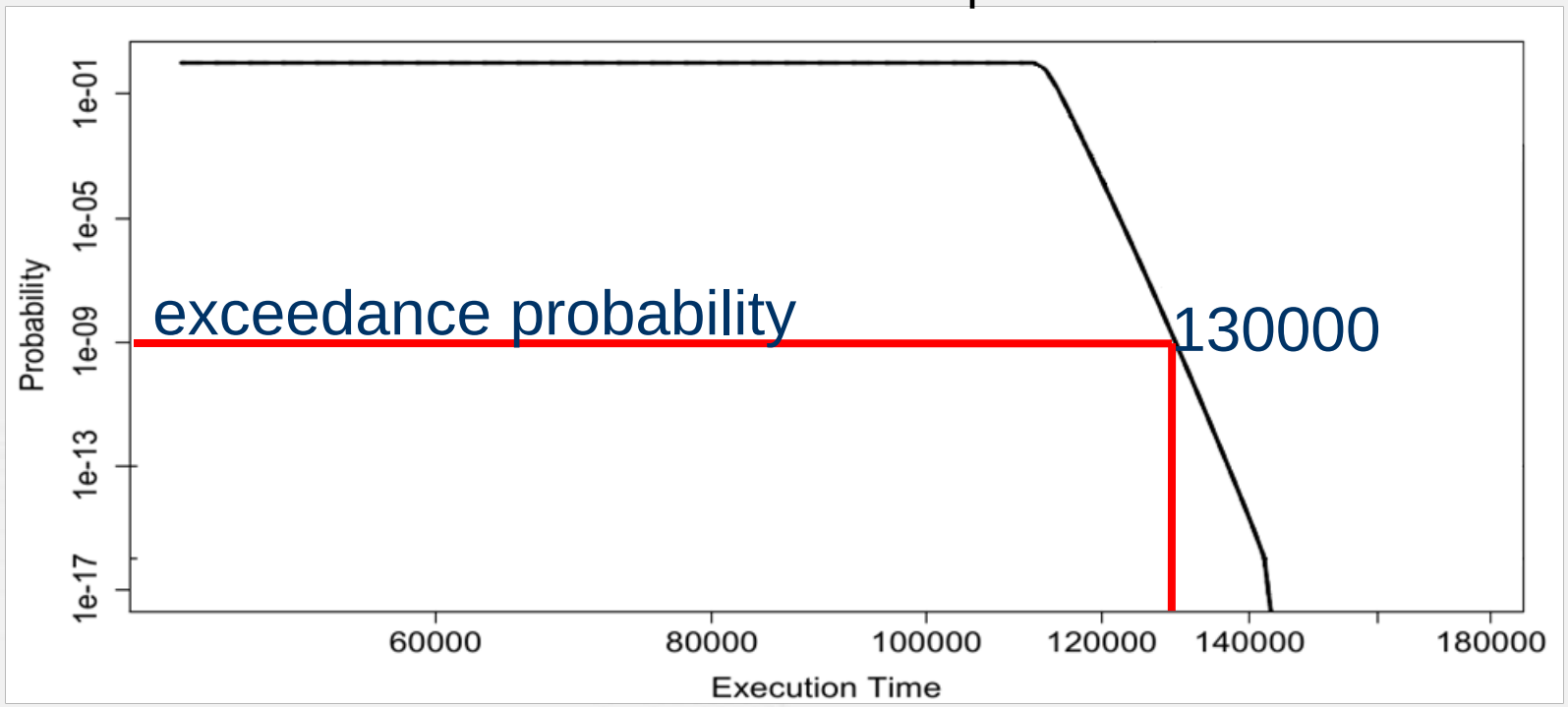
Probabilistic Timing Analysis Approach

- *Reduce dependence on execution history*
  - HW and SW whose execution time behaviour does not depend on execution history
    - While benefiting performance-improving hardware!

- *How:*
  - Introduce **randomisation** into the **timing behaviour** of the **hardware** and software
    - The functional behaviour is left unchanged

Probabilistic Timing Analysis

- *PTA allows cutting the WCET bound tail at the level of probability suited for the system (e.g. 10^{-16} per hour of operation)*
  - Prob. Failure per hour = probability of failure of the program **x** execution rate per hour



Measurement-Based PTA

- *Execution times of end-to-end runs to be modeled by independent and identically distributed (i.i.d.) random variables*
  - Independence: occurrence of one event doesn't affect the occurrence of the other event
  - Identical distribution: same probability distribution
- *Events that affect execution time to be random events*
- *Existence of the probability is enough, no need to be computed [1]*

[1] L. Cucu et al, Measurement-Based Probabilistic Timing Analysis for Multi-path Programs, *ECRTS 2012*

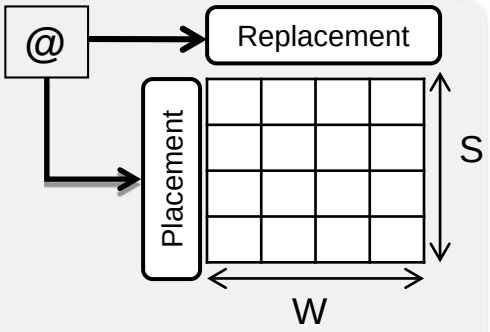
MBPTA requirements on cache

- *For every instruction we have an Execution Time Profile:*
  - ETP = $\{I_1, I_2, \ldots, I_n\}\{p_1, p_2, \ldots, p_n\}$
  - $\sum p_i = 1$
- *Having an ETP for each instruction ensures MBPTA can be applied [1]*
- *For the case of the cache we are interested in:*
  - all events affecting the latency of a cache access
  - the probability of each event
  - the value of the latency itself
- *Example of single level cache cache:*
  - $\{I_{hit}, I_{miss}\}\{p_{hit}, p_{miss}\}$

[1] L. Cucu et al, Measurement-Based Probabilistic Timing Analysis for Multi-path Programs, *ECRTS 2012*

Time-Randomised Single level Caches

- *Set-associative cache*
- *Remove sources of determinism*
  - Placement
  - Replacement
- *How?*
  - Random placement
  - Random replacement



P_{miss} can be approximated:

Sequence: $A_i\ B_{i+1}\ \ldots\ B_{j-1}\ A_j$, with B_i accessing different cache lines

$$P_{miss_{A_j}}(S,W) = \left(1 - \left(\frac{W-1}{W}\right)^{\sum_{l=1}^{l=k} P_{miss_{B_l}}}\right) \cdot \left(1 - \left(\frac{S-1}{S}\right)^{k}\right)$$

Multilevel Cache concepts

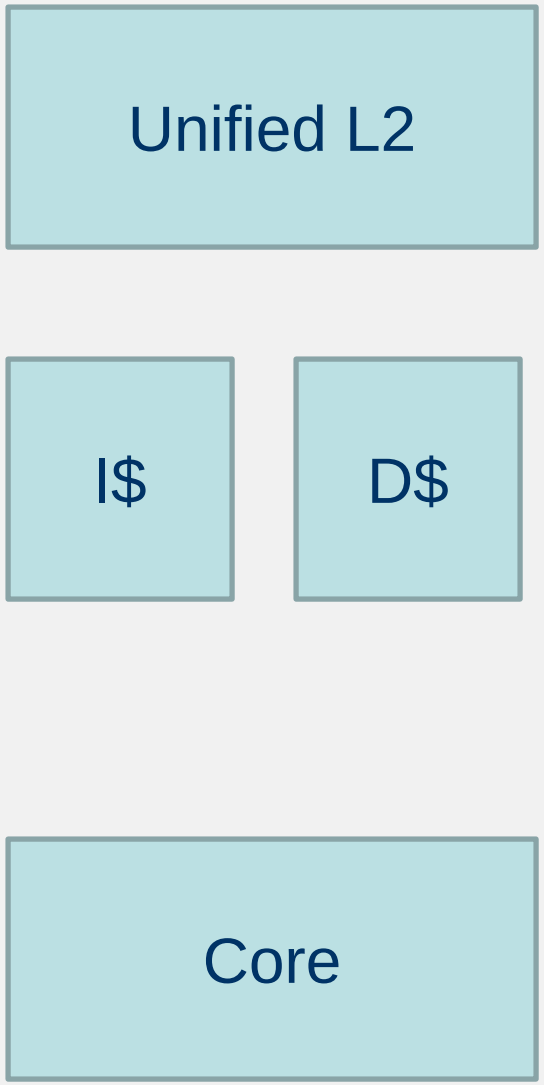
- *In multilevel caches many events affect the latency/probability of an access*
- *Inclusivity*
  - Inclusive caches
  - Exclusive caches
  - Non-Inclusive caches
- *Write policy*
  - Write-through
  - Write-back
- *Write-allocation policy*
  - Write Allocate
  - No Write Allocate
- *Unified/Shared Caches*

Outline

- Motivation
- Background
  - Probabilistic Timing Analysis
    - Measurement-Basted Timing Analysis requirements on HW
  - Time-Randomised Caches
  - Cache concepts: inclusivity, write-miss and allocation policies
- MBPTA for multilevel caches
  - NIC, WT-nWA caches
- Generalisation of the model
- Results
- Conclusions

Time-Randomised Multi-level Cache

- *Considered organisation:*
  - Fully associative caches with random replacement policy
    - it also works for set-associative, see our paper
  - Separate first-level Instruction and Data Caches
    - Write-through
    - Write-no allocate
  - Unified non-inclusive L2
    - Write-back
    - Write-allocate



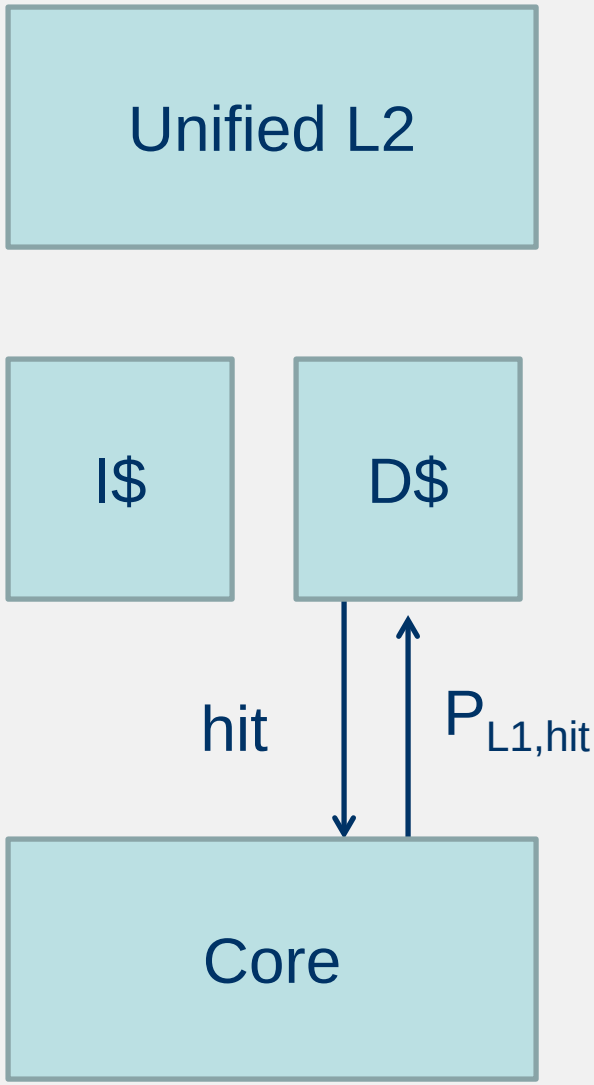
L1 access

- *Event Probability: $P_{L1,hit}$*
- *$P_{L1,hit} = 1 - P_{L1,miss}$*

$$P_{miss_{A_j}}(W) = \left(1 - \left(\frac{W-1}{W}\right)^{\sum_{l=1}^{l=k} P_{miss_{B_l}}}\right)$$

- *Base gives the probability of eviction on every miss*
- *Exponent gives an idea of the number of misses*

The probabilities are approximations.
MBPTA does not require to compute them



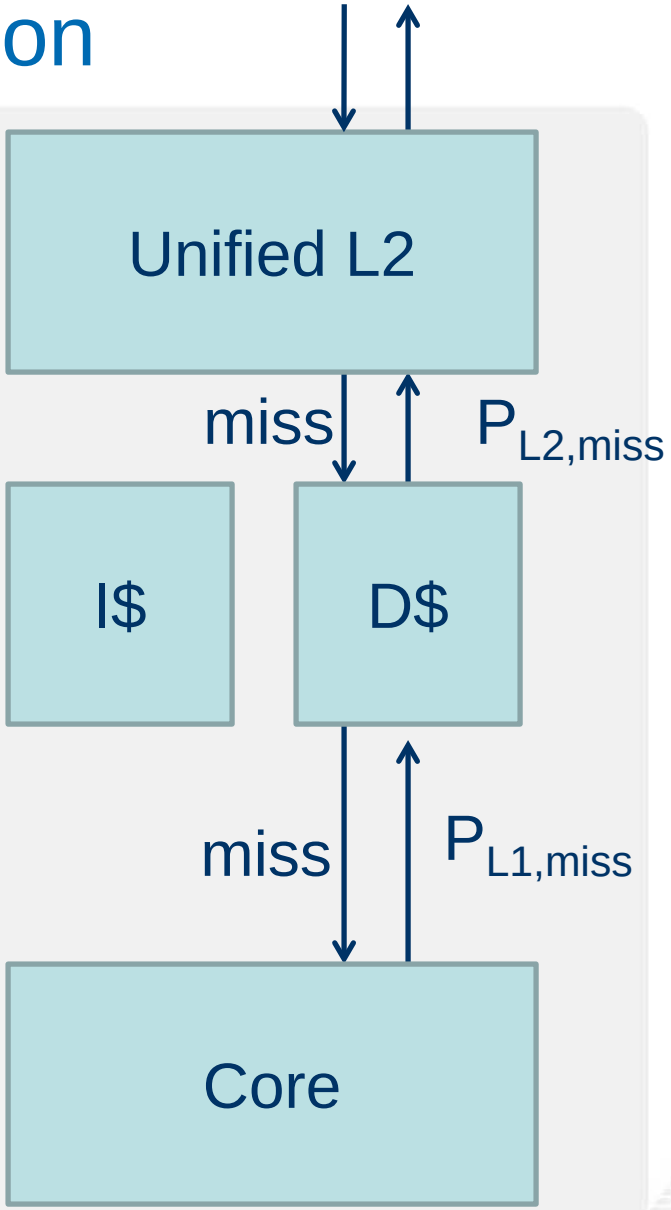
L2: Impact of DL1 Configuration

- *Considering DL1 configuration*
  - Write-through + write-no-allocate
  - Write-back + write-allocate
- *Unified non-inclusive L2*
  - Write-back + write-allocate

$$P_{UL2,miss}(@_A) = 1 - \left(\frac{W_{UL2} - 1}{W_{UL2}}\right)^{\sum_{i=1}^{k} P_{UL2,miss}(L2acc_i)}$$

WT+WNA:
IL1 misses + DL1 LD misses + **DL1 ST all**

WB+WA:
IL1 misses + DL1 LD misses + **DL1 ST misses**



Generalisation of the model

- *Accesses to the same cache lines introduce probabilistic dependences*
  - Actual probability computation more complex
  - No need to be computed, probabilistic nature is enough for MBPTA
- *Different cache line sizes*
- *Inclusivity*
  - Adds additional evictions
    - Just one more probabilistic event
- *Several Levels of cache*
  - Additional probabilistic events

Outline

- Motivation
- Background
  - Probabilistic Timing Analysis
    - Measurement-Basted Timing Analysis requirements on HW
  - Time-Randomised Caches
  - Cache concepts: inclusivity, write-miss and allocation policies
- MBPTA for multilevel caches
  - NIC, WT-nWA caches
- Generalisation of the model
- Results
- Conclusions

Experimental Setup

- *Single core pipelined processor similar to LEON4*
- *Time-randomised caches with random placement and replacement*
  - Instruction L1: 4KB, 4-way set-associative, 1 cycle hit latency
  - Data L1: 4KB, 4-way set-associative, 1 cycle hit latency
  - L2 configurations:
    - Unified: 128 KB, 8-way set-associative, 10 cycles hit latency
    - Write-Back Write Allocate

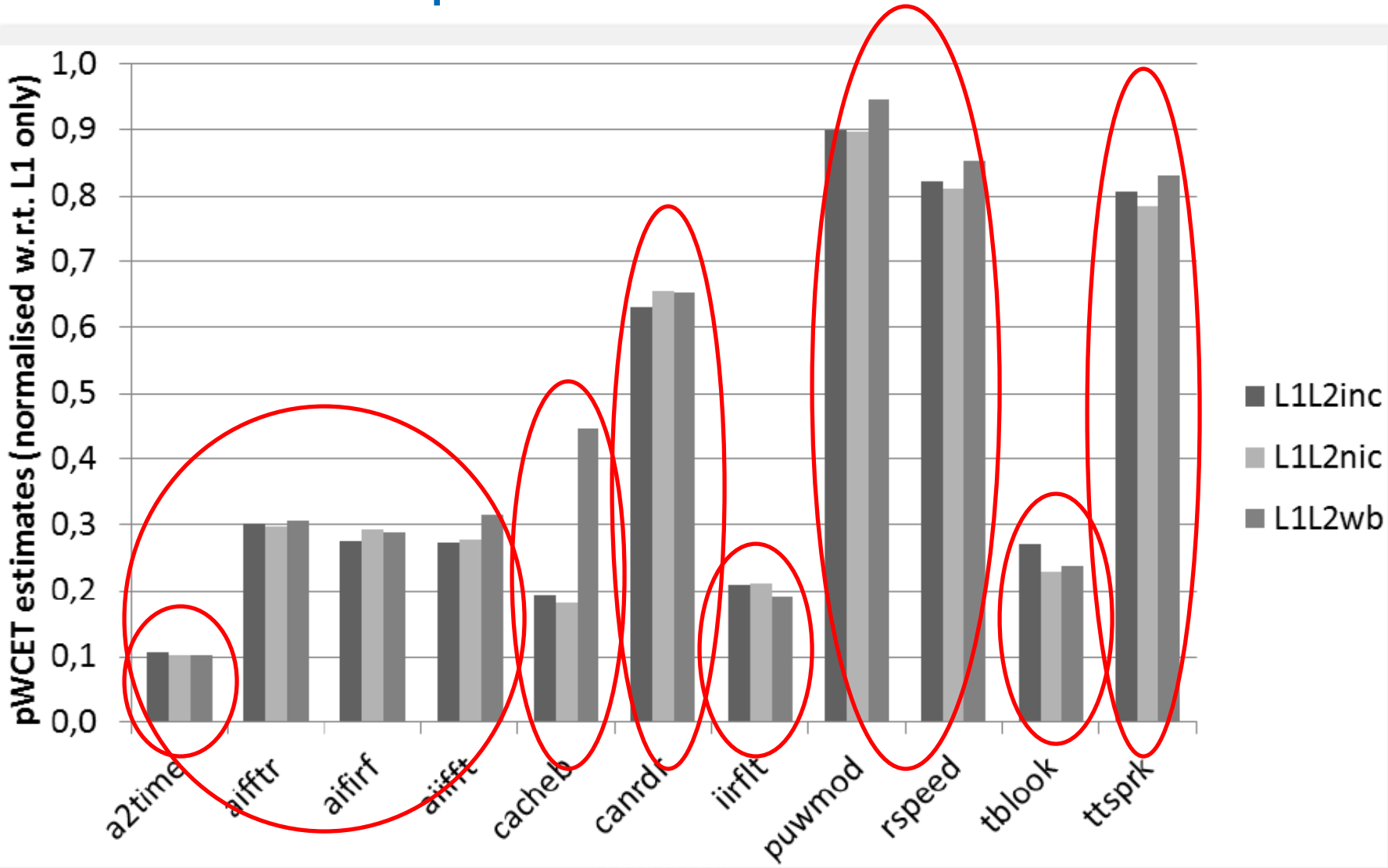
Configuration	DL1 Write Policy	DL1 Write Allocation	Inclusivity
L1-L2 INC	Write-through	No-allocate	Inclusive
L1-L2 NIC	Write-through	No-allocate	Non-inclusive
L1-L2 WB	Write-Back	Write-allocate	Inclusive

Results of i.i.d. tests

- *Pipeline model similar to LEON4*
- *Various cache configurations (inclusivity/write policies)*

Benchmarks	L1-L2 INC	L1-L2 NIC	L1-L2 WB	L1 (only)
a2time	0.03/0.29	0.83/0.41	0.46/0.44	0.90/0.49
aifftr	0.71/0.74	0.95/0.33	0.82/0.59	1.19/0.33
aifirf	0.40/0.11	1.04/0.20	0.13/0.94	1.04/0.79
aiifft	0.68/0.32	0.50/0.41	0.96/0.17	1.09/0.94
cacheb	0.63/0.93	1.11/0.72	1.20/0.35	0.79/0.66
canrdr	0.79/0.16	0.75/0.54	1.00/0.37	0.32/0.91
iirflt	0.96/0.85	0.68/0.41	0.78/0.50	0.07/0.22
puwmod	1.39/0.67	0.99/0.25	0.94/0.75	0.30/0.71
rspeed	0.47/0.43	1.33/0.51	0.91/0.24	1.35/0.42
tblook	1.33/0.92	0.52/0.86	0.34/0.26	0.76/0.44
ttsprk	0.19/0.43	0.89/0.52	0.21/0.42	0.67/0.63

Reduction in pWCET estimates



Conclusion

- *Measurement-based Probabilistic Timing Analysis combined with time-randomised caches are a cost-effective solution for industry*
- *Multi-level Time-randomised caches are also MBPTA-compliant*
  - Unified caches
  - Different inclusivity/write allocation/write policies
  - Arbitrary levels
- *55% pWCET reduction on average*