EPC: Extended Path Coverage for Measurement-based Probabilistic Timing Analysis

M. Ziccardi†, E. Mezzetti† and T. Vardanega†
J. Abella§ and F.J. Cazorla§‡

†University of Padua
§BSC
‡Spanish National Research Council

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° 611085

www.proxima-project.eu
Overview of MBPTA

- Probabilistic execution time estimation
  - Based on measurements
  - Relies on solid probabilistic and statistical basis
    - Posing statistical requirements on the input data (i.i.d.)
  - Estimates are attached a *probability of exceedance* (user-provided threshold)
Overview of MBPTA

- **Probabilistic execution time estimation**
  - Based on measurements
  - Relies on solid probabilistic and statistical basis
    - Posing statistical requirements on the input data (i.i.d.)
  - Estimates are attached a *probability of exceedance* (user-provided threshold)
Overview of MBPTA

- Probabilistic execution time estimation
  - Based on measurements
  - Relies on solid probabilistic and statistical basis
    - Posing statistical requirements on the input data (i.i.d.)
  - Estimates are attached a probability of exceedance (user-provided threshold)
Overview of MBPTA

- Probabilistic execution time estimation
  - Based on measurements
  - Relies on solid probabilistic and statistical basis
    - Posing statistical requirements on the input data (i.i.d.)
  - Estimates are attached a *probability of exceedance* (user-provided threshold)

![Diagram showing MBPTA process](image_url)
Overview of MBPTA

- Probabilistic execution time estimation
  - Based on measurements
  - Relies on solid probabilistic and statistical basis
    - Posing statistical requirements on the input data (i.i.d.)
  - Estimates are attached a probability of exceedance (user-provided threshold)
Overview of MBPTA

- Probabilistic execution time estimation
  - Based on measurements
  - Relies on solid probabilistic and statistical basis
    - Posing statistical requirements on the input data (i.i.d.)
  - Estimates are attached a *probability of exceedance* (user-provided threshold)
Overview of MBPTA

- Probabilistic execution time estimation
  - Based on measurements
  - Relies on solid probabilistic and statistical basis
    - Posing statistical requirements on the input data (i.i.d.)
  - Estimates are attached a probability of exceedance (user-provided threshold)
Representativeness of observations

- **Inherent limitation of measurement-based approaches**
  - Bounds are only valid for the set of paths and execution conditions for which observations were collected

- **The same applies to MBPTA**
  - Probabilistically captures variability from history of execution...
  - ...but results are only valid for the subset of observed paths
Representativeness of observations

- **Inherent limitation of measurement-based approaches**
  - Bounds are only valid for the set of paths and execution conditions for which observations were collected

- **The same applies to MBPTA**
  - Probabilistically captures variability from history of execution...
  - ...but results are only valid for the subset of observed paths
Extending the path coverage

- Synthetically extending the set observed paths

Measurements collected over a subset of the program paths

+ Synthetic measurements for unobserved paths

"Fully representative" distribution for all paths in a program obtained from both observed AND synthetic observations

Representativeness gap

Distribution valid only for observed paths

Execution time

Exceedance Probability

No EPC
EPC building blocks

- **Path-independence**
  - Execution times (ET) can be made independent from the path through which they have been collected.
  - EPC exploits *probabilistic path independence* at *basic blocks* level.

- **Synthetic measurements over unobserved paths**
  - Path-independent ET can be combined to construct representative execution times for end-to-end (unobserved) paths.

- **Cannot naively sum up the maximum observed ET!**
  - Collected observations are only relative to a particular path.
  - Includes cache-level and core-level dependencies.
Probabilistic path independence

- **Path-independent execution times for a basic block**
  - Values does not depend on a particular path
  - Summing up a penalty or padding to each observed ET to compensate for any positive effect due to a specific path (e.g., cache behavior)

- **Example of deterministic independence (caches)**
  - Assume all accesses were hit and add a miss-hit latency to each observed value
  - For each memory access in a basic block:
    \[
    Obs^+(bb_i) = Obs(bb_i, \phi) + \sum_{@I \in bb_i} pad^I(@I) + \sum_{@D \in bb_i} pad^D(@D)
    \]
  - This is way overly pessimistic!

- **Exploit the probabilistic framework**
  - No need to enforce each observation to upper-bound the worst-case behavior
ATPs and probabilistic padding

- **On time-randomized single-core architectures**
  - Randomized caches are the main source of variability
  - $ATP(A, \phi) = \begin{pmatrix} \frac{L_{hit}}{P_{hit}(A, \phi)} & L_{miss} \\ \frac{P_{hit}(A, \phi)}{P_{miss}(A, \phi)} & \frac{P_{miss}(A, \phi)}{P_{miss}(A, \phi)} \end{pmatrix}$

- **Probabilistic padding of ATPs**
  - Adding a probabilistic padding to negatively compensate potential positive effects of variability (e.g., a cache hit) on a specific path
  - $\overline{ATP}(A) = \begin{pmatrix} \frac{L_{hit}}{P_{hit}(A)} & \frac{L_{miss}}{P_{miss}(A)} \\ \frac{P_{hit}(A)}{P_{miss}(A)} & \frac{P_{miss}(A)}{P_{miss}(A)} \end{pmatrix}$

- **Computing the padding probability**
  - Ensure the resulting ATP distribution follows the worst ET distribution for that basic block (for any program path)
  - Cannot modify a set of observations to follow the exact distribution of the worst-case $\overline{ATP}$s
    - Would require to selectively compensate the effects of cache hits on a subset of the collected observations
Over-approximating the worst-case ATP

\[ \text{ATP}(\lnot A, \phi) \]

\[ \text{ATP}(\lnot A) \]

\[ \text{ATP}^+(\lnot A) \]

\[ L_{\text{hit}} \quad L_{\text{miss}} \quad L_{\text{miss+pad}} \]
Over-approximating the worst-case ATP

\[ \text{ATP}(\@A, \phi) \] 

\[ \overline{\text{ATP}}(\@A) \] 

\[ \text{ATP}^+(\@A) \]

Diagram showing the ATP, Cumulative ATP, \( L_{\text{hit}} \), \( L_{\text{miss}} \), \( L_{\text{miss+pad}} \).
Over-approximating the worst-case ATP

ATP(@A, φ)  ATP(@A)  ATP+(@A)
Over-approximating the worst-case ATP

\[ \text{ATP}(A, \varphi) \quad \text{ATP}_+(A) \quad \text{ATP}(A) \]

\[ L_{\text{hit}} \quad L_{\text{miss}} \quad L_{\text{miss+pad}} \]

Cumulative ATP

\[ L_{\text{hit}} \quad L_{\text{miss}} \quad L_{\text{miss+pad}} \]
Computing the padding probability

Formulation of padding probability on $ATP^+$

$$ATP^+(A) = ATP(A, \phi) \otimes \begin{pmatrix} 0 & L_{pad} \\ 1 - P_{pad}(A, \phi) & P_{pad}(A, \phi) \end{pmatrix}$$

$$= \begin{pmatrix} L_{hit} & L_{miss} \\ P_{hit}(A, \phi) & P_{miss}(A, \phi) \end{pmatrix} \otimes \begin{pmatrix} 0 & L_{pad} \\ 1 - P_{pad}(A, \phi) & P_{pad}(A, \phi) \end{pmatrix}$$

$$= \begin{pmatrix} L_{hit} & L_{miss} & L_{miss} + L_{pad} \\ P_{hit}^+(A) & P_{miss}^+(A) & P_{miss+pad}^+(A) \end{pmatrix}$$
Computing the padding probability/2

- Under one single requirement

\[ ATP^+(@A) \geq \overline{ATP}(@A) \]

implying

\[ P^+_{hit}(@A) \leq \overline{P_{hit}}(@A) \]
\[ P_{hit}(@A) + P^+_{miss}(@A) \leq \overline{P_{hit}}(@A) + \overline{P_{miss}}(@A) \]

- Lower bound to \( P_{pad}(@A, \phi) \)

\[ P^+_{hit}(@A) \leq \overline{P_{hit}}(@A) \]
\[ P_{hit}(@A, \phi) \cdot (1 - P_{pad}(@A, \phi)) \leq \overline{P_{hit}}(@A) \]
\[ \ldots \]
\[ P_{pad}(@A, \phi) \geq 1 - \frac{\overline{P_{hit}}(@A)}{\overline{P_{hit}}(@A, \phi)} \]
Computing the padding probability/3

**Definition (Reuse distance - \textit{rd})**

The \textit{reuse distance} of $\alpha_A$ on a path $\phi$ is defined as the number of memory blocks mapped to the same set of $\alpha_A$ accessed between $\alpha_A$ and the previous access to the memory block containing $\alpha_A$.

**Definition (Unique accesses - \textit{un})**

With \textit{unique accesses}, instead, we refer to the number of distinct memory blocks mapped to the same set of $\alpha_A$ accessed in between $\alpha_A$ and the previous access to the memory block containing $\alpha_A$ on a path $\phi$.

\[
P_{hit}(\alpha_A) = 1 - P_{miss}(\alpha_A) \quad \text{where} \quad P_{miss}(\alpha_A) = \begin{cases} 
1 - \left( \frac{w-1}{w} \right)^{rd(\alpha_A)} & \text{if } rd(\alpha_A) < w \\
1 & \text{otherwise}
\end{cases}
\]

\[
P_{hit}(\alpha_A, \phi) \leq uP_{hit}(\alpha_A, \phi) = \begin{cases} 
\frac{1}{(\frac{w-1}{w})^{un(\alpha_A,\phi)-w+1}} & \text{if } un(\alpha_A,\phi) < w \\
\text{otherwise}
\end{cases}
\]
Application of padding and synthetic paths

- **Computational complexity in** $P_{pad}$ **relatively low**
  - Both $rd$ and $un$ computed on a restricted scope (only immediate predecessors)

- **Collected observations of basic blocks updated**
  - This affects both observed values and frequencies

- **Synthetic path generation**
  - Collect artificial execution times by iterating over all the basic blocks in each unobserved path
  - Consider only a subset of all possible paths
    - Pruning based on known flow facts
Feeding data into MBPTA

Measurements collected over a subset of the program paths

Synthetic measurements for unobserved paths

"Fully representative" distribution for all paths in a program obtained from both observed AND synthetic observations

Representativeness gap

Distribution valid only for observed paths

No EPC

Exceedance Probability

Execution time

Trace collection

SETV

Randomized HW platform

HW Tracing Support

MBPTA

EVT

Block selection

Convergence criteria

Curve fitting

Tail extension

MBPTA

Convergence criteria

Block selection

Curve fitting

Tail extension

Hardware Tracing Support

Synthetic measurements for unobserved paths

HW Tracing Support

Trace collection

SETV

Randomized HW platform

"Fully representative" distribution for all paths in a program obtained from both observed AND synthetic observations

Representativeness gap

Distribution valid only for observed paths

No EPC

Exceedance Probability

Execution time
Evaluation

- **EPC extends MBPTA**
  - Representativeness of full path coverage → *at what cost?*
  - Evaluate the compromise between tightness and representativeness
  - Against both
    - Base MBPTA (from ECRTS-12)
    - Path Upper Bounding (PUB) approach (from ECRTS-14)
      - Based on balancing of conditional branches
      - Modified executable used at analysis time

- **On PROXIMA simulator**
  - SoCLib-based cycle-level platform simulator
  - Single level *time-randomized* I and D caches
    - 4-ways set associative
    - Random placement and replacement policies
Against standard MBPTA

Normalized pWCET (at probability 10^-15)

1.45
1.4
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1

fac recursion insertsort edn matmult fcntl fibcall jfclint ns prime lcdnm bs cnt janne fir cover adpcm crc aiffr cacheb irifl aiiff matrix aiiff rspeed a2time tblock puwmod Malardalen EEMBC Auto

Single Path

Known Worst-Case Path (WCP)

Malardalen

No WCP

EEMBC Automotive

Avg.
From simulation to real implementation

- **Implementation within the PROXIMA project**
  - Randomized platform under finalization
  - EPC is being implemented on top of industrial-quality tool
  - Tracing requirements fulfilled by available HW tracing support
Conclusions

- **Presented an hybrid approach**
  - Extends and complements MBPTA
  - Attack the problem of path representativeness

- **Showed fully-representative results can be had**
  - Computationally feasible
  - Incurs limited pessimism in the results as compared to current state-of-the-art approaches

- **EPC moving from prototyping to implementation**
  - On top of real HW and industrial-quality tool