A Framework For The Evaluation Of Measurement-based Timing Analyses

Benjamin Lesage, David Griffin, Frank Soboczenski, Iain Bate, Rob Davis

RTNS 2015 - November 3rd
**Context**

*pWCET* estimation

- **pWCET**: Bound the occurrence of timing events in the system
  - WCET with attached exceedance probability
- **Sound**: Upper-bound the actual execution time
- **Tight**: Close to the actual execution time
Context
pWCET estimation

- **pWCET**: Bound the occurrence of timing events in the system
  - WCET with attached exceedance probability
- **Sound**: Upper-bound the actual execution time
- **Tight**: Close to the actual execution time
Context

- **pWCET**: Bound the occurrence of timing events in the system
  - WCET with attached exceedance probability
- **Sound**: Upper-bound the actual execution time
- **Tight**: Close to the actual execution time
Context
pWCET estimation

- **Sound** and **Tight** comparisons are difficult without a ground truth
  - Smaller estimates may be optimistic
  - Larger estimates may be pessimistic
Context

MBPTA – Measurement Based Probabilistic Timing Analysis

- **MBPTA**: derive a pWCET from runs of the analysed task
  - Predicts the tail of the pWCET using Extreme Value Theory

- Abstraction from the analysed platform and task
  - Sources of execution time variability must be bounded
  - Analysed samples must cover all paths in the application

[ECRTS 2012]
Context

MBPTA – Measurement Based Probabilistic Timing Analysis  [ECRTS 2012]

- **MBPTA**: derive a pWCET from runs of the analysed task
  - Predicts the tail of the pWCET using Extreme Value Theory

- Abstraction from the analysed platform and task
  - Sources of execution time variability must be bounded
  - Analysed samples must cover all paths in the application
Context

MBPTA – Measurement Based Probabilistic Timing Analysis  [ECRTS 2012]

- **MBPTA**: derive a pWCET from runs of the analysed task
  - Predicts the tail of the pWCET using Extreme Value Theory
- Abstraction from the analysed platform and task
  - Sources of execution time variability must be bounded
  - **Analysed samples must cover all paths in the application**
Framework for the evaluation of MBPTA

Overview

- **Program model**: Generate the structure of a task
  - Configured by the end-user

- **Temporal model**: Attach temporal information to blocks
  - Relies on **Basic Block Measurements**
  - Abstractions allow exact pWCET computation

- **Model simulator**: Collection of time samples
  - Controlled to satisfy coverage requirements
  - Samples fed to the Timing analysis
Outline

- Context
- Framework for the evaluation of MBPTA
  - Temporal Model
  - Task
  - pWCET
  - BBM
- Evaluation
- Conclusion
Framework for the evaluation of MBPTA

Independent Block Model

- **Basic block**: sequence of instructions with a single entry/exit

- The behaviour of a block depends on the **platform P** and its **state s**

\[ P(\text{b}, s) = (t', s') \]
Framework for the evaluation of MBPTA

Independent Block Model

- **Basic block**: sequence of instructions with a single entry/exit

- The behaviour of a block depends on the **platform $P$** and its state $s$

  \[ P(b, s) = (t', s') \]

- Sources of execution time variability must be bounded
  - Through **probabilistic** or deterministic mechanisms
  - Contributes to the independence of blocks’ behaviour

- Focus on path coverage requirement
Framework for the evaluation of MBPTA

Independent Block Model

- Basic block: sequence of instructions with a single entry/exit

- The behaviour of a block depends on the platform $P$
  - Captured by an Execution Time Profile: $ETP_b$
  - Independent of the execution history
  - Akin to the output of low-level timing analyses

- Path: a finite sequence of basic blocks

  $\pi : b_1 \rightarrow b_2 \rightarrow b_3 \rightarrow b_4$

  - The execution time of a path is the convolution of its components

  $pET(\pi) = \bigotimes_{b \in \pi} ETP_b$
Framework for the evaluation of MBPTA
Representing tasks

- **Task**: a finite set of paths
  - Represented as an **Abstract Syntax Tree** (AST)
  - Tree nodes map to syntactic structures in code
  - Leafs map to basic blocks in code

- Capture standard programming patterns
- Ease reasoning about WCET computation

- No arbitrary flow between blocks
- No support for flow constraints
Framework for the evaluation of MBPTA

Generating tasks

- Start from the root of the tree
- Randomly pick node type
  - Selection constrained by user
- Generate relevant node type parameters
- Generate subtree for all node children
Framework for the evaluation of MBPTA

Generating tasks

- Start from the root of the tree
- Randomly pick node type
  - Selection constrained by user
- Generate relevant node type parameters
- Generate subtree for all node children
Framework for the evaluation of MBPTA

Generating tasks

- Start from the root of the tree
- Randomly pick node type
  - Selection constrained by user
- Generate relevant node type parameters
- Generate subtree for all node children
Framework for the evaluation of MBPTA

Generating tasks

- Start from the root of the tree
- Randomly pick node type
  - Selection constrained by user
- Generate relevant node type parameters
- Generate subtree for all node children
Framework for the evaluation of MBPTA

Generating tasks

- Start from the root of the tree
- Randomly pick node type
  - Selection constrained by user
- Generate relevant node type parameters
- Generate subtree for all node children
Start from the root of the tree

Randomly pick node type
  - Selection constrained by user

Generate relevant node type parameters

Generate subtree for all node children
The behaviour of a node is independent of the execution history
- Both in timings and execution path
Framework for the evaluation of MBPTA

pWCET computation – An example

- The behaviour of a node is independent of the execution history
  - Both in timings and execution path
Framework for the evaluation of MBPTA

pWCET computation – An example

- The behaviour of a node is independent of the execution history
  - Both in timings and execution path

pWCET(BODY) does not change across iterations.
(Path execution time does.)
Framework for the evaluation of MBPTA

pWCET computation – An example

- The behaviour of a node is independent of the execution history
  - Both in timings and execution path

\[ ETP_{BODY} = pWCET(BODY) \]
Framework for the evaluation of MBPTA

\[ \text{pWCET computation} – \text{An example} \]

- The behaviour of a node is independent of the execution history
  - Both in timings and execution path

\[ ETP_{\text{COND}} = \text{pWCET(COND)} \]
Framework for the evaluation of MBPTA

The behaviour of a node is independent of the execution history
- Both in timings and execution path

The pWCET of a node is a combination of its children
- Similar to tree-based WCET computation
- Relies on convolution (⊗) and envelope (∪) operations

\[ pWCET(LOOP) = pWCET(COND)^{iter+1} \otimes pWCET(BODY)^{iter} \]
Framework for the evaluation of MBPTA
Gathering basic block measurements (BBM)

- Capture timings off a real application
  - Ensure representative low level timings
  - Assume independence of blocks
  - Assume covering observations

- Extract the structure of the application
  - Valgrind Instrumentation framework
  - Extract traces of memory accesses

- Collect cache hits/misses at the block level
  - Callgrind instrumentation tool
  - Simulate a randomised memory hierarchy
  - Satisfy architectural requirements of MBPTA
  - Capture probabilistic profiles

- Instrument FFmpeg h264 decoding primitive
  - Readily available input vectors
  - Vast array of basic block profiles
Evaluation
Realism – Experimental conditions

Does the framework produces realistic execution time traces?

- Compare observed and simulated execution times

  - **Observed**: Collect execution time and path for each run
    - Build BBM of blocks across all runs
    - Process \( \approx 8000 \) frames per input vector

  - **Simulated**: Simulate each observed path in the framework
    - Pick execution times in traversed BBM
    - Ignore dependencies between traversed blocks

- Input vectors from the archive.org movie database
Evaluation
Realism - PLAN

Exceedance Probability

Execution Time (MCycles)

- Observed
- Simulated
Tight fit between observed and simulated distributions
Evaluation
Realism - PLAN

Loss of precision at low probabilities.
Evaluation

Realism - PLAN

"Moments" of the original distribution kept
Evaluation
Realism - NOSF

![Graph showing Exceedance Probability versus Execution Time (MCycles) for Observed and Simulated data. The graph plots a downward curve, indicating a decrease in Exceedance Probability as Execution Time increases. The Observed data is represented by a solid blue line, while the Simulated data is shown with a dashed red line.](image-url)
Evaluation

Realism - NOSF

Exceedance Probability

Execution Time (MCycles)

Keep overall shape of the distribution

Observed

Simulated
Evaluation

Realism - NOSF

Exceedance Probability

Execution Time (MCycles)

Keep overall shape of the distribution
Evaluation

Robustness - Experimental conditions

How robust is MBPTA in the absence of path coverage?

- Compare predicted and exact pWCET
  - Metric: Normalised pWCET at $10^{-9}$ (over exact value)

- Control coverage of samples fed to the analysis
  - Enforce path coverage during simulations
  - Randomly ban nodes in the AST
  - Only ban non-dominating nodes

- 100 randomly generated tasks
  - Pick ETP in BBM database
  - 8000 runs per sample
  - Remove tasks with un-coverable path set
  - 2 samples per task/per experiment
Evaluation
Robustness - Results

Normalised pWCET (10-9)

Omitted blocks

5% 50% 90%
Evaluation
Robustness - Results

Consistent analysis results Even with omitted blocks
Evaluation
Robustness - Results

Few compliant tasks in the later scenarios

Normalised pWCET (10-9)

Omitted blocks

5%
50%
90%

39
Evaluation

Robustness - Results

Normalised $pWCET$ (10-9)

Missing leafs covered by Same-depth alternatives
Evaluation
Robustness - Results

Normalised pWCET (10^-9)

Omitted nodes

5%
50%
90%
Evaluation

Robustness - Results

Degraded precision and tightness of the analysis
Evaluation

Robustness - Results

Normalised pWCET (10^-9)

Whole subset of the tree left unseen
Conclusion

A framework for measurement-based timing analyses:
- Abstract the superfluous from the platform model
- Rely on observed timing data
- Build upon existing high-level timing analyses
- Detect problems, not their absence

On the robustness of MBPTA:
- Path coverage is an expensive requirement
- Biased samples can produce sound estimates

Future work
- Introduce (controlled) dependencies between blocks
- Introduce (controlled) dependencies between runs
Questions ?
- Post’it: alegri / 4freephotos.com
- Abascus: HB / Wikimedia.org
- Torn paper: http://imgarcade.com
Evaluation

Realism - KUNG

Exceedance Probability

Execution Time (MCycles)

- Observed
- Simulated
Evaluation

Realism - KUNG

Exceedance Probability

Execution Time (MCycles)
Evaluation
Realism - PHOP

Exceedance Probability

Execution Time (MCycles)

- Observed
- Simulated
Evaluation
Realism - PHOP

Exceedance Probability

Execution Time (MCycles)