Measurement-Based Probabilistic Timing Analysis in the presence of Buffer Resources

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

13th WCET Workshop.
July 9, 2013, Paris, France

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.
Motivation

- **Critical Real-Time Embedded (CRTE) systems**
  - Used in Space, Aerospace, Transportation,… industries
  - More functional value → more computational power → more complex HW/SW

- **Complex HW/SW affect time analysability**
  - Makes it difficult for current timing analysis to compute trustworthy and tight WCET estimates

- **Probabilistic Timing Analysis (PTA)**
  - Not as mature as more conventional timing analysis techniques
  - Aims at enabling/simplifying the analysis of complex processor architectures
Current architectures use EH to boost average-case performance by speculating on the future

- E.g.: temporal and spatial locality for caches

**Static timing analysis requires EH information**

- For example to determine whether a given memory access hits in cache STA needs to built
  - The full history of all access to the cache to keep the cache state current after every access → the more abstraction the more pessimism
Background: Controlling EH

• **There are increasing limitations in acquiring EH info**
  - Complex processor architectures (IP protected)
  - Incomplete and/or inaccurate documentation
  - Program information may be unknown at analysis time

• **Reduction of available knowledge about HW/SW → pessimistic assumptions → degradation of the tightness of the WCET**

- Design HW and SW whose execution time behavior does not depend on EH
  - Without renouncing performance-accelerator HW features
  - Without affecting functional behavior

- Develop new timing analysis techniques to go with it
**Background: PTA**

**Approach:**
- Introduces *randomisation* into the *timing behaviour* of the hardware and software
  - The functional behaviour is left unchanged
- Provides new *probabilistic timing analysis* techniques

**Want to know more?**
- PROARTIS and PROXIMA
PROXIMA IP Project Consortium

- **Roles**
  - Academic research (BSC, UPD, INR, UoY)
  - Tool chain providers at hardware (AG), software (SYS) and timing analysis (RPT) levels;
  - Industrial exploitation via major EU companies and technology centres in the avionics, space, automotive, and railway sectors (AIF, AST, IFX, IKR)
ETP: Execution Time Profile

• In a PTA-conformant architecture:
  - The timing behaviour of instructions can be represented with an Execution Time Profile (ETP).
  - ETP is the probability distribution function describing the different execution times that an instruction can take.

\[
(l, p) = \{l_1, l_2, \ldots, l_k\}\{p_1, p_2, \ldots, p_k\} \sum_{i=1}^{k} p_i = 1
\]

• Do I have to change entirely my processor design?
  - ONLY certain hardware resources are randomised to achieve that behavior, e.g. cache [1] OR software-support for randomisation is used instead [2]
Measurement-based PTA (MBPTA)

- Complete runs of the program are made on the time-randomised target platform
  - Measurements **must capture** outcome of events that make execution time vary
    - All potential execution times must have a probability of being exercised
  - The ETP of processor instructions need **not** be known
    - Existence of ETP → guarantees ET act as a die with an arbitrarily large number of faces, each representing a distinct execution time
  - From this and sufficiently large number of i.i.d observations (in the order of hundreds) we can use EVT to derive pWCET estimations

**Measurement-Based Probabilistic Timing Analysis for Multi-path Programs**
ECRTS 2012
Complete runs of the program are made on the time-randomised target platform

- Measurements must capture outcome of events that make execution time vary
  - All potential execution times must have a probability of being exercised
- The ETP of processor instructions need not be known
  - Existence of ETP → guarantees ET act as a die with an arbitrarily large number of faces, each representing a distinct execution time
- From this and sufficiently large number of i.i.d observations (in the order of hundreds) we can use EVT to derive pWCET estimations

Simple Processor Resources

• **The existence of ETP ensures i.i.d behavior**
  - ETP is the implementation of a random variable
  - How is this obtained?

• **Jitter:**
  - Difference between the best & worst latency of any resource
  - Due to the history of requests or the data contained in request

• **Jitterless resources**
  - No response time variation

• **Jittery resources**
  - Variable impact on the WCET whose significance depends
    - The program under study
    - The analysis method
Dealing with jittery resources (no timing anomalies)

- Force that the requests to the resources incur the worst-case latency.
  - Acceptable if the cumulative impact on the WCET is deemed low enough by the system designer
- Worst latency is not acceptable $\rightarrow$ randomise the timing behaviour of the resource (e.g. caches)
Resources $\rightarrow$ instructions

- This simple resource classification does not cover more complex resource such as buffers

Instructions
- Access multiple resources that can be arranged in different manners, e.g. sequentially or in parallel.
- Under each arrangement, the ETP of those resources can be properly combined to derive the ETP of the instruction.
- Convolution
Two stages (fetch and execute) that respectively access time randomised caches\(^1\): IL1 and DL1.

- Probability of hit for every access

In between both stages there is a 2-entry buffer

- \(i_1\) may introduce delays on \(i_2\)–\(i_4\) if it misses in DL1 (IL1 hit prob. is 1.0)
- \(i_2\)–\(i_4\) make no data cache access

---


Chronograms: \(<DL1-i1, IL1-i2, IL1-i3>\)

\(<H,H,H> - P(HHH) = 0.378\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No buffer stalls

\(<H,M,H> - P(HMH) = 0.162\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No buffer stalls

\(<M,H,H> - P(MHH) = 0.042\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(i4: 6\) cycles of buffer stall

\(<H,H,M> - P(HHM) = 0.252\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No buffer stalls

\(<H,M,M> - P(HMM) = 0.108\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No buffer stalls

\(<M,H,M> - P(MHM) = 0.028\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(i4: 4\) cycles of buffer stall

\(<M,M,H> - P(MHM) = 0.018\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(i4: 4\) cycles of buffer stall

\(<M,M,M> - P(MMM) = 0.012\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i2</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i3</td>
<td>F</td>
<td>F</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i4</td>
<td>F</td>
<td>b</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(i4: 2\) cycles of buffer stall
Observation

- Given a set of fixed (or probabilistic\(^1\)) initial conditions
  - Each combination of prob. events (e.g. DL1 and IL1 accesses) → one fully-deterministic behaviour of the buffer.
- Buffer introduces a different number of stall cycles for each combination of probabilistic events.
  - For a sequence of random events the behaviour of the buffer is fully deterministic
  - All data dependences, which are given by the sequence of instructions that are executed and their order.
- MBPTA works or a per-path basis → in each path the sequence of instructions executed is known and fixed across runs of the same path.
- Buffers do not introduce new probabilistic events!
  Increase the ‘latency’ under some prob. event outcome

---

Probability Tree
Probability Tree

<table>
<thead>
<tr>
<th>Instruction id</th>
<th>Instruction type</th>
<th>IL1 hit prob.</th>
<th>IL1 miss prob.</th>
<th>DL1 hit prob.</th>
<th>DL1 miss prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>LD</td>
<td>1.0</td>
<td>0.0</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>i2</td>
<td>ADD</td>
<td>0.7</td>
<td>0.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>i3</td>
<td>ADD</td>
<td>0.6</td>
<td>0.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>i4</td>
<td>ADD</td>
<td>1.0</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
New Classification for jittery resources/1

• (i) History dependence
  - Whether the jitter is produced solely by the event under consideration (no history dependence) or
  - The combination of previous events and the current one (history dependence);
  - History Dependent (HD) or not History Dependent (nHD)

• (ii) Type of jitter
  - whether the jitter is deterministic (DJ)
  - probabilistic (PJ) or
  - simply propagated/transmitted regardless of its source (TJ)
New Classification for jittery resources

- **noHD + DJ**
  - Resource’s latency does not depend on the sequence of requests it has received, but on the data of each request.
  - Some FP unit in is affected by the particular operands
    - We force it to experience worst latency
    - Worst-case mode [1].

- **noHD + PJ**
  - We do not have any particular realistic example of this type

- **noHD + TJ**
  - In principle such a resource cannot exist because it does not produce any jitter by itself and cannot propagate any jitter if it is history independent

---

New Classification for jittery resources/3

- HD + PJ
  - *Time randomised cache*
  - The sequence of events between two consecutive accesses to the same data together with the initial cache state, determine the hit/miss probability of that access.
    - Time randomised caches have been shown to be analysable with MBPTA [1].

- HD + DJ
  - Deterministic cache (e.g. modulo+LRU)
  - Events may experience different latencies depending on history
    - Not analysable by MBPTA unless the factors that influence jitter are fully under control

---

• HD+ TJ
  - This is the case of a hardware buffer.
  - An instruction may spend a different number of cycles in a buffer depending on previous events.
  - Propagate deterministically the effect of the jitter induced by other resources.
    - If such jitter is probabilistic, then the stalls induced by buffers occur also with a given probability and so they are
Conclusions

• Buffer resources
  ➢ Do not create any jitter on their own
  ➢ They propagate inbound jitter regardless of the nature of it.
  ➢ Do not introduce new probabilistic events! Increase the ‘latency’ of some prob. events
  ➢ Do not break PTA requirements,
  ➢ Can be used in PTA-conforming processors with no change.

• Provide a comprehensive classification of hardware resources and how they can be considered in the context of PTA.
Measurement-Based Probabilistic Timing Analysis in the presence of Buffer Resources

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

13th WCET Workshop.
July 9, 2013, Paris, France

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.