Static probabilistic WCET Estimation for architectures with Faulty Instruction Caches

Damien Hardy, Isabelle Puaut (firstname.lastname@irisa.fr)
Motivation

• **Real-time systems** are subject to **timing constraints**
  – All tasks have to meet their deadlines
Motivation

• **Real-time systems** are subject to **timing constraints**
  – All tasks have to meet their deadlines

• **Temporal validation**
  – Estimation of **Worst-Case Execution Time (WCET)**
Motivation

- **Real-time systems** are subject to **timing constraints**
  - All tasks have to meet their deadlines

- **Temporal validation**
  - Estimation of **Worst-Case Execution Time (WCET)**
Motivation

• **Real-time systems** are subject to **timing constraints**
  – All tasks have to meet their deadlines

• **Temporal validation**
  – Estimation of **Worst-Case Execution Time (WCET)**
WCET estimation
WCET estimation

• Measurement $\Rightarrow$ unsafe
  – Execution + Timing measurement
  – Hard to execute tasks in their worst-case scenario
WCET estimation

- **Measurement** => **unsafe**
  - Execution + Timing measurement
  - Hard to execute tasks in their worst-case scenario

- **Static analysis** => **safety ensured**
  - Analysis of the program structure
  - WCET estimation
    - Low level analysis: modeling of hardware timing
    - High level analysis: worst-case execution path
WCET estimation

- **Measurement => unsafe**
  - Execution + Timing measurement
  - Hard to execute tasks in their worst-case scenario

- **Static analysis => safety ensured**
  - Analysis of the program structure
  - WCET estimation
    - Low level analysis: modeling of hardware timing
    - High level analysis: worst-case execution path
  - **Assumption: the hardware is fault-free**
However...

Tomorrow failures will not be exceptional
However...

Tomorrow failures will not be exceptional

• Making hardware fault-free not scalable
  – Increase spares, larger cells => more area, cost, power

A Resilience Roadmap [DATE10]
However...

Tomorrow failures will not be exceptional

- Making hardware fault-free not scalable
  - Increase spares, larger cells => more area, cost, power

- We cannot expect free for aging lifetime
However...

Tomorrow failures will not be exceptional

- Making hardware fault-free not scalable
  - Increase spares, larger cells => more area, cost, power

- We cannot expect free for aging lifetime

Solution: Disabling faulty entries
Permanent faults & WCET estimation

• **Disabling faulty entries**
  – For caches => less capacity => **additional cache misses**
    • Significant average performance degradation [MICRO’12]
Permanent faults & WCET estimation

• **Disabling faulty entries**
  – For caches => less capacity => **additional cache misses**
    • Significant average performance degradation [MICRO’12]

• **Static WCET estimation**
  => **unsafe in the presence of faults**

---

[Image of a person standing in front of a chalkboard with a graph on it.]
Contribution

Static probabilistic WCET estimation
- Static: to capture all scenarios (path)
- Probabilistic: to capture the effect of faults
  - Random location and random number of faults

WCET probability distribution
Assumptions

**Architecture**
- Set-associative instruction cache
  - LRU replacement policy

**Fault model**
- Permanent Faulty SRAM cells (bits)
  - Equal probability of failure: $p_{fail}$
  - Random location
- A faulty cache block is disabled
  - At least one bit faulty

$w = \#\text{ways}$
$s = \#\text{sets}$

TAG+ECC  \[\times\]  DATA+ECC

$k$ bits
Some Probabilities

- **Cache Block faulty**
  \[ p_{bf} = 1 - (1 - p_{fail})^k \]

- **To have \( i \) ways faulty in a set**
  - Binomial probability law
  \[ p_{wf}(i) = \binom{w}{i} (p_{bf})^i (1 - p_{bf})^{w-i} \]
void main(){
    int i,j,k;
    i++; j++; k++; ...
}
Background: static WCET estimation (1/2)
Background: static WCET estimation (1/2)

• Cache Analysis: abstract interpretation
  - Guaranteed hit at runtime
  - Always hit/First miss

\[
\begin{array}{c|c|c}
 & \text{MRU} & \text{LRU} \\
\hline
\text{SET 0} & \{a\} & \emptyset \\
\text{SET 1} & \emptyset & \{b\} \\
\text{SET 2} & \emptyset & \emptyset \\
\text{SET 3} & \emptyset & \{c\} \\
\end{array}
\]
Background: static WCET estimation (1/2)

- **Cache Analysis: abstract interpretation**
  - Guaranteed hit at runtime
  - Always hit/First miss

- **Implicit Path Enumeration Technique (IPET)**
  - Maximum flow problem
  - Linear programming solver

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>{a}</td>
<td>{}</td>
</tr>
<tr>
<td>SET 1</td>
<td>{}</td>
<td>{b}</td>
</tr>
<tr>
<td>SET 2</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>SET 3</td>
<td>{}</td>
<td>{c}</td>
</tr>
</tbody>
</table>
Background: static WCET estimation (2/2)

- **Outputs**
  - Worst case execution time (WCET)
  - Worst case execution path (WCEP)
  - Cache hit behavior: #hits per set & LRU position along the WCEP
Penalty estimation
Penalty estimation

• First step: single path program
Penalty estimation

• **First step: single path program**

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP
Penalty estimation

- **First step: single path program**

<table>
<thead>
<tr>
<th>SET</th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

Faulty cache configuration:

```
0
1
2
0
0
```

#hits per set & LRU position along the WCEP
Penalty estimation

- **First step: single path program**

<table>
<thead>
<tr>
<th>SET 0</th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>120</td>
<td>10 X</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14 X</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP

Faulty cache configuration

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
Penalty estimation

- **First step: single path program**

<table>
<thead>
<tr>
<th>SET 0</th>
<th>SET 1</th>
<th>SET 2</th>
<th>SET 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRU</td>
<td>120</td>
<td>150</td>
<td>180</td>
</tr>
<tr>
<td>LRU</td>
<td>10</td>
<td>14</td>
<td>13</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP

Faulty cache configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Fault-induced misses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>164</td>
</tr>
</tbody>
</table>

=> 10 fault-induced misses
=> 164 fault-induced misses
Penalty estimation

• First step: single path program

<table>
<thead>
<tr>
<th>SET</th>
<th>#hits MRU</th>
<th>#hits LRU</th>
<th>Faulty cache configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP

Penalty = 174 x Tmiss

=> 10 fault-induced misses

=> 164 fault-induced misses
Penalty estimation

• First step: single path program

<table>
<thead>
<tr>
<th>SET</th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

Faulty cache configuration

- Fault-induced misses: 10
- Fault-induced misses: 164

Penalty = 174 x Tmiss

Probability = P_{wf}(1) x P_{wf}(2) x P_{wf}(0) x P_{wf}(0)
Penalty distribution estimation

• First step: single path program

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP
Penalty distribution estimation

- First step: single path program
- Property: sets are independent

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP
Penalty distribution estimation

• First step: single path program
• Property: sets are independent
  – Probability distribution per set

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP
Penalty distribution estimation

- First step: single path program
- Property: sets are independent
  - Probability distribution per set

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP
Penalty distribution estimation

- **First step**: single path program
- **Property**: sets are independent
  - Probability distribution per set
  - Convolution

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>

#hits per set & LRU position along the WCEP
Permanent Faults impact WCEP
Permanent Faults impact WCEP

- Fault-free Estimation

![Diagram showing fault-free estimation with nodes A, B, C, and D, and time values $T_B=100$ and $T_C=90$.]
Permanent Faults impact WCEP

- Fault-free Estimation
  - WCEP: A B D

T_B=100
T_C=90
Permanet Faults impact WCEP

- Fault-free Estimation
  - WCEP: A B D
- A Fault affects $T_C$

$T_B = 100$
$T_C = 90 + 100$
Permanent Faults impact WCEP

- Fault-free Estimation
  - WCEP: A B D
- A Fault affects $T_C$
  - $T_C > T_B$
Permanent Faults impact WCEP

- **Fault-free Estimation**
  - **WCEP**: A B D

- **A Fault affects** $T_c$
  - $T_c > T_B$
  - **WCEP**: A C D
Permanent Faults impact WCEP

- Fault-free Estimation
  - WCEP: A B D
- A Fault affects $T_C$
  - $T_C > T_B$
  - WCEP: A C D

WCEP Variation $\Rightarrow$ #hits along the WCEP cannot be used
Penalty distribution estimation

- **Exhaustive WCET computation**
  - \#faulty cache configuration = \((w+1)^s\)
Penalty distribution estimation

- **Exhaustive WCET computation**
  - \#faulty cache configuration = \((w+1)^s\)

- **Estimation of the number of accesses detected as hit**
  - Upper bound for each set and each LRU position
    - Insensitive to path variation
  - Modified IPET (see paper)
    - \(s \times w\) ILP resolutions

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>
Penalty distribution estimation

- Using an upper bound of the number of accesses detected as hit
  - Ensure to never underestimate the penalty
  - But it may be pessimistic
    - Mutually exclusive accesses to different sets are all considered

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>
Penalty distribution estimation

- **Using an upper bound of the number of accesses detected as hit**
  - Ensure to never underestimate the penalty
  - But it may be pessimistic
    - Mutually exclusive accesses to different sets are all considered

- **Tightness improvement**
  - Focus cases: sets entirely faulty
    - Post convolution treatment
    - Worst-case penalty estimation
      - ILP formulation (see paper)
      - s ILP resolutions

<table>
<thead>
<tr>
<th></th>
<th>MRU</th>
<th>LRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 0</td>
<td>120</td>
<td>10</td>
</tr>
<tr>
<td>SET 1</td>
<td>150</td>
<td>14</td>
</tr>
<tr>
<td>SET 2</td>
<td>180</td>
<td>13</td>
</tr>
<tr>
<td>SET 3</td>
<td>220</td>
<td>20</td>
</tr>
</tbody>
</table>
Experimental setup

• Analyzed codes: 25 benchmarks
  • From the Mälardalen WCET benchmark suite
  • single path/multi-path programs (up to 43KB)
  • loop intensive/control programs

• Metric
  • Accuracy: Exhaustive vs. Base and Improved

• Instruction cache configuration (pfail=10^{-4})
  • 1KB cache, Direct-mapped & 2-way set associative
  • 64B cache line, SEC-DED ECC code,
  • Latency: 1/100 cycles

• Software: Heptane and Cplex 12.5
Accuracy: no WCEP variation

nsichneu (SetAssoc ; pfail=1e-04)

Probability (1-CDF) vs. WCET (in cycles)

- Exhaustive
- Base
- Improved
Accuracy: no WCEP variation

nsichneu (SetAssoc ; pfail=1e-04)

Equal to the exhaustive (10/25 Benchmarks)
Accuracy: if-then-else outside loops

adpcm (SetAssoc; pfail=1e-04)

Exhaustive
Base
Improved
Accuracy: if-then-else outside loops

adpcm (SetAssoc ; pfail=1e-04)

Very close to the exhaustive (6/25 Benchmarks)
Accuracy: if-then-else inside loops

statemate (SetAssoc ; pfail=1e-04)

Exhaustive
Base
Improved
Accuracy: if-then-else inside loops

Probability (1-CDF) vs. WCET (in cycles)

- Exhaustive
- Base
- Improved

Close to the exhaustive (9/25 Benchmarks)
Faults: Significant impact on WCET
Exploration of architectural parameters
Exploration of architectural parameters

Different cache associativity

jfdctint (pfail=1e-04)
Exploration of architectural parameters

Different cache associativity

The higher the associativity, the lower the impact on WCET
Exploration of architectural parameters

Different cache block size (4-way cache)

Probability (1-CDF) vs. WCET (in cycles) for jfdctint (pfail=1e-04)

- 64B
- 32B
- 16B
Exploration of architectural parameters

Different cache block size (4-way cache)

Smaller blocks can be profitable when considering the impact of permanent faults
Conclusion & Future work

• Method to calculate a probabilistic WCET bound
  – Static analysis: to always detect the longest execution path
  – Probabilistic: to capture the effect of faulty blocks
  – Accurate and computationally tractable
  – Allows hardware parameters exploration to mitigate the impact of faults
Conclusion & Future work

• **Method to calculate a probabilistic WCET bound**
  – Static analysis: to always detect the longest execution path
  – Probabilistic: to capture the effect of faulty blocks
  – Accurate and computationally tractable
  – Allows hardware parameters exploration to mitigate the impact of faults

• **Future work**
  – Faults in other hardware components
  – Interaction between faulty components
  – Fault management mechanism exploration
Thank you! Questions?