Modeling of a Bimodal Branch Predictor for WCET Analysis using the IPET Method

Claire Burguière, Christine Rochange
IRIT-UPS, Toulouse
Introduction

- Worst-Case Execution Time
- Static Methods: IPET
- Predictability:
  - Low cost of the analysis
  - WCET tight
- Branch Prediction Modeling:
  - Estimating the maximum number of mispredictions
  - Integrating misprediction penalties in WCET analysis
Outline

- Branch Prediction Modeling
- Impact of Branch Prediction
- Model Description
- Results
- Branch Prediction Modeling
- Impact of Branch Prediction
- Model Description
- Results
Branch Prediction

branch instruction @

BHT

If taken, pc=

BTB

If not taken, pc=

prediction

target @

next @
Static Analysis

- Ignoring aliasing
  - BHT indexed by the instruction address:
    - Analysis of the counter assigned to this branch
  - Indexed using histories:
    - Analysis of all path leading to this branch ($2^n$ counter)

- Including aliasing
  - Determining all branches sharing the same entry
  - Global Analysis of all branches
Previous Models

- Static simulation [ColinPuaut2000]
  - Timing schema (algorithmic structure)
- IPET method [MitraRoychoudhury2002]
  - Set of constraints describing execution (CFG)
    - Significant number of constraints
    - Dynamic predictor using global histories
      ⇒ no predictable [Engblom2003]
- Ignoring aliasing [BateReutemann2004]
  - Upper-bound on the number of mispredictions
    - More complex algorithmic structures [BuRo2005]
Preventing Aliasing

- Modeling aliasing:
  - Complex or not tight
    - No predictable

- Preventing aliasing:
  - Mixed branch predictor [PaEm00]:
    - One entry assigned to each branch predicted dynamically
    - Other branch: static prediction
  - Intel IA-64
    - "branch hints" assigned by the compiler
  - Selecting *predictable* branch
- Branch Prediction Modeling
- Impact of Branch Prediction
- Model Description
- Results
Misprediction Penalty

\[
\text{Penalty} = \text{Execution time with misprediction} - \text{Execution time with "good-prediction"}
\]

- Previous Penalty:
  - Global penalty
  - Per-branch penalty
Penalty Distribution

misprediction penalty (cycles)

# cond. branches
Penalty Variation

Difference between penalties measured on each direction (taken, not-taken)
Impact of Branch Prediction

- **Per-direction penalty**
  - For each conditional branch $B$
    - Modifications of the CFG
    - $wp = ”well-predicted”$
    - $mp = mispredicted$
- Branch Prediction Modeling
- Impact of Branch Prediction
- Model Description
- Results
int i, j, fact, somme3=0,sommef=0;
for (i=0 ; i>N ; i++)
{
    fact=1;
    for (j=2 ; j<tab[i] ; i++)
        fact=fact×j;
    if (tab[i]%3 ==0)
    {
        tab[i]=tab[i]/3;
        som3=som3+tab[i];
    }
    else somf=somf+fact;
}
Structural Constraints

\[
\begin{align*}
&b_0 = 1 = a_{0,1} \\
&b_1 = a_{0,1} + a_{7,1} = a_{1,8} + a_{1,9} + m_{1,8} + m_{1,9} \\
&b_2 = a_{9,2} + a_{3,2} = a_{2,4} + a_{2,3} + m_{2,4} + m_{2,3} \\
&b_3 = a_{2,3} + m_{2,3} = a_{3,2} \\
&b_4 = a_{2,4} + m_{2,4} = a_{4,5} + a_{4,6} + m_{4,5} + m_{4,6} \\
&b_5 = a_{4,5} + m_{4,5} = a_{5,7} \\
&b_6 = a_{4,6} + m_{4,6} = a_{6,7} \\
&b_7 = a_{5,7} + a_{6,7} = a_{7,1} \\
&b_9 = a_{1,9} + m_{1,9} = a_{9,2} \\
&b_8 = a_{1,8} + m_{1,8} = 1
\end{align*}
\]
Flow Constraints
and Overall Execution Time

\[ T = \sum_{0 \leq i \leq 9} b_i \times t_{b_i} + \sum_{(i,k) \in A} a_{i,k} \times t_{a_{i,k}} + \sum_{(i,k) \in C} m_{i,k} \times p_{i,k} \]

- \( m_{i,k} \) = misprediction number of the edge \( i \rightarrow k \)
- \( p_{i,k} \) = execution time of the branch \( i \rightarrow k \) when it is mispredicted
- \( A \) = set of edges
- \( C \) = set of conditional branch edges

\[ b_7 \leq 20 \]
\[ b_2 \leq 100 \]
Branch Prediction Modeling for bloc 2

- Past counter state:

\[ b_2 = b_2^{00} + b_2^{01} + b_2^{10} + b_2^{11} \]
\[ m_2 = m_2^{00} + m_2^{01} + m_2^{10} + m_2^{11} \]
\[ m_2^{00} \leq b_2^{00} \]
\[ m_2^{01} \leq b_2^{01} \]
\[ m_2^{10} \leq b_2^{10} \]
\[ m_2^{11} \leq b_2^{11} \]
Branch Prediction Modeling for bloc 2 (2)

- Repeating counter state:
  - for each counter state $c$:
    $b_2^c = p_{0\rightarrow 2}^c + p_{2\rightarrow 2}^c + p_{2\rightarrow 8}^c$
  - Initial and final state:
    $\sum_c p_{0\rightarrow 2}^c = 1$
    $\sum_c p_{2\rightarrow 8}^c = 1$
  - for each $c$:
    $\sum_c p_{2\rightarrow 2}^{c,0} + \sum_c p_{2\rightarrow 8}^{c,0} = a_{2,3} + m_{2,3}$
    $\sum_c p_{2\rightarrow 2}^{c,1} + \sum_c p_{2\rightarrow 8}^{c,1} = a_{2,4} + m_{2,4}$
    $p_{2\rightarrow 8}^c = p_{2\rightarrow 8}^{c,0} + p_{2\rightarrow 8}^{c,1}$
Branch Prediction Modeling for bloc 2 (3)

- Counter Evolution
  - Reaching the state:
    \[ b_2^{00} = p_{2\rightarrow 2}^{00,0} + p_{2\rightarrow 2}^{01,0} + p_{0\rightarrow 2}^{00} \]
    \[ b_2^{01} = p_{2\rightarrow 2}^{00,1} + p_{2\rightarrow 2}^{10,0} + p_{0\rightarrow 2}^{01} \]
    \[ b_2^{10} = p_{2\rightarrow 2}^{01,1} + p_{2\rightarrow 2}^{11,0} + p_{0\rightarrow 2}^{10} \]
    \[ b_2^{11} = p_{2\rightarrow 2}^{10,1} + p_{2\rightarrow 2}^{11,1} + p_{0\rightarrow 2}^{11} \]
  - Leaving the state:
    \[ b_2^c = p_{2\rightarrow 2}^c + p_{2\rightarrow 2}^c + p_{2\rightarrow 8}^c \]
Branch Prediction Modeling for bloc 2 (4)

- Mispredictions
  - Occurrences:
    \[ m_{2}^{00} = p_{2\Rightarrow 2}^{00,1} + p_{2\Rightarrow 8}^{00,1} \]
    \[ m_{2}^{01} = p_{2\Rightarrow 2}^{01,1} + p_{2\Rightarrow 8}^{01,1} \]
    \[ m_{2}^{10} = p_{2\Rightarrow 2}^{10,0} + p_{2\Rightarrow 8}^{10,0} \]
    \[ m_{2}^{11} = p_{2\Rightarrow 2}^{11,0} + p_{2\Rightarrow 8}^{11,0} \]

  - Linked to mispredictions edges:
    \[ m_{2,3} = m_{2}^{10} + m_{2}^{11} \]
    \[ m_{2,4} = m_{2}^{00} + m_{2}^{01} \]
- Branch Prediction Modeling
- Impact of Branch Prediction
- Model Description
- Results
### Penalty

<table>
<thead>
<tr>
<th>PENALTY</th>
<th>WCET (cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>2831</td>
</tr>
<tr>
<td>Per-branch</td>
<td>2609</td>
</tr>
<tr>
<td>Per-direction</td>
<td>2557</td>
</tr>
</tbody>
</table>
In Respect of Modeling Aliasing

<table>
<thead>
<tr>
<th></th>
<th>gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of constraints</td>
<td>50%</td>
</tr>
<tr>
<td>Computation time</td>
<td>33%</td>
</tr>
</tbody>
</table>
Conclusion and Future Work

- Preventing aliasing:
  - Prediction mode for each branch
    - *Intel IA-64*
- Misprediction penalty
  - Per-direction penalty
- Branch prediction modeling
  - Optimized binary code
- Future work:
  - branch classification: static or dynamic
THANKS!
QUESTIONS?