The O-GEHL branch predictor

Optimized GEometric History Length

André Seznec
IRISA/INRIA/HIPEAC
What is classic 😊

- Global history based:
  - Yeh and Patt 91, Pan and So 91

- Use of multiple history lengths:
  - McFarling 93, Evers et al. 96

- Use an adder tree instead of a meta-predictor
  - Vintan and Iridon 99, Jiménez and Lin 01
Multiple history length neural predictor
The O-GEHL branch predictor

GEometric History Length predictor

The set of history lengths forms a geometric series

\[
\begin{align*}
L(0) &= 0 \\
L(i) &= \alpha^{i-1} L(1)
\end{align*}
\]

\{0, 2, 4, 8, 16, 32, 64, 128\}

What is important: \(L(i) - L(i-1)\) is drastically increasing
Updating the predictor

- Update on misprediction and under a threshold

8-bit counters and perceptron update threshold (29)

→ Would not have qualified for CBP-1 😞
Dynamic update threshold fitting

On an O-GEHL predictor, best threshold depends on:
- the application
- the predictor size
- the counter width

By chance for the best fixed threshold,
updates on mispredictions ≈ updates on correct predictions

Monitor both numbers and adapt the update threshold
8 components 8 bits counter would qualify for CBP-1
Counter width on O-GEHL predictors

- 8 bits are just overkilling 😞
- 4 bits are sufficient 😊
- **Mixing 5 bits for short histories and 4 bits for long histories is slightly better 😊**
- 3 bits are not so bad !!
Adaptative history length fitting

*(inspired by Juan et al 98)*

\[
\begin{align*}
&\left(\frac{1}{2} \text{ applications: } L(7) < 50\right) \\
&\neq \\
&\left(\frac{1}{2} \text{ applications: } L(7) > 150\right)
\end{align*}
\]

Let us adapt some history lengths to the behavior of each application

- 8 tables:
  - T2: L(2) and L(8)
  - T4: L(4) and L(9)
  - T6: L(6) and L(10)
Adaptative history length fitting (2)

Intuition:
- if high degree of aliasing on T7, stick with short history

Implementation:
- monitoring of aliasing on updates on T7 through a tag bit and a counter

Simple is sufficient:
- Flipping from short to long histories and vice-versa
The O-GEHL branch predictor

André Seznec
Caps Team
Irisa
Information to be hashed

- Address + conditional branch history:
  - path confusion on short histories 😞
- Address + path:
  - Direct hashing leads to path confusion 😞

1. Represent all branches in branch history
2. Use also path history (1 bit per branch, limited to 16 bits)
Configuration for CBP

- 8 tables:
  - 2 Kentries except T1, 1Kentries
  - 5 bit counters for T0 and T1, 4 bit counters otherwise
  - 1 Kbits of one bit tags associated with T7

\[ 10K + 5K + 6 \times 8K + 1K = 64K \]

- \( L(1) = 3 \) and \( L(10) = 200 \)
  - \( \{0, 3, 5, 8, 12, 19, 31, 49, 75, 125, 200\} \)
Hashing 200+ bits for indexing !!

- Need to compute 11 bits indexes:
  - Full hashing is unrealistic

1. Just regularly pick at most 33 bits in:
   - address + branch history + path history
2. A single 3-entry exclusive-OR stage
A case for the OGEHL predictor (1)

- High accuracy

- Robustness to variations of history lengths choices:
  - \( L(1) \) in \([2,6]\), \( L(10) \) in \([125,300]\)
  - misp. rate < 1.04 x reference misp. rate

- Geometric series: not a bad formula!!
  - best geometric \( L(1)=3, L(10)=223 \), REF-0.02 misp/KI
  - best overall \{0, 2, 4, 9, 12, 18, 31, 54, 114, 145, 266\}
    REF-0.04 misp/KI
A case for the OGEHL predictor (2)

- Reduce counter width by 1 bit: 49 Kbits
  ➔ would have been a finalist 😊

- 64 Kbits 4 components OGEHL predictor
  ➔ would have been a finalist 😊

- 50 Kbits 4 components OGEHL predictor (3-bit)
  ➔ would have been a finalist 😊

- 768 Kbits 12 components OGEHL predictor
  ➔ 2.25 misp/KI
A case for the O-GEHL predictor (3)

- O-GEHL predictor uses only global information
- Can be ahead pipelined
- Prediction computation logic complexity is low

(The End)