Adaptive Information Processing: An Effective Way to Improve Perceptron Predictors

Hongliang Gao    Huiyang Zhou

Computer Science Department
University of Central Florida
Information system model

- Information system model by Chen et. al. [ASPLOS-VII].
- Key observations
  - Shortcomings:
    - Fixed information vector while different workloads/branches need different information data.
  - Perceptron weights ↔ Correlation
    - Assemble information vector to maximize correlation
- Our contribution
  - Re-assemble the information vector based on correlation (weights)
  - Performed at a coarse grain, so it is not latency critical
Adaptive Information Processing

- Profile-directed adaptation
- Correlation-directed adaptation
Profile-directed adaptation

MAC-RHSP predictor [5]
Profile-directed adaptation

<table>
<thead>
<tr>
<th>Type</th>
<th>FP</th>
<th>INT</th>
<th>MM</th>
<th>SERV</th>
</tr>
</thead>
</table>

**Table 1**

<table>
<thead>
<tr>
<th>FP</th>
<th>INT</th>
<th>MM</th>
<th>SERV</th>
</tr>
</thead>
<tbody>
<tr>
<td>P[0:3]</td>
<td>L[0:3]</td>
<td>G[0:3]</td>
<td>L[0:3]</td>
</tr>
</tbody>
</table>

**Table 2**

<table>
<thead>
<tr>
<th>FP</th>
<th>INT</th>
<th>MM</th>
<th>SERV</th>
</tr>
</thead>
<tbody>
<tr>
<td>P[0:3]</td>
<td>L[0:3]</td>
<td>G[0:3]</td>
<td>L[0:3]</td>
</tr>
</tbody>
</table>

**Table N**

University of Central Florida
**Workload Detector**

- **Detection criteria**
  - **SERV:** a large number of static branches
  - **FP:** a small number of static branches, a high number of floating point operation, and a high number of instructions using XMM registers
  - **MM:** a medium number of static branches, a medium number of floating point operation, and a medium number of instructions using XMM registers
  - **INT:** default
Correlation-directed adaptation

Information feeding logic

LHR table
GHR
PC

MUX1
MUX2

c1
c2
cN

wt table 1
wt table 2
wt table N

Index 0
Index 1
Index 2
Index N

Index 1
Index 2
Index N

Adaptation Logic

(c1, c2, c3, ...)

Prediction = sign(y)

MAC-RHSP predictor
Correlation-directed adaptation

Information

- LHR table
- GHR
- PC

Information feeding logic

Table 1
- type (INT)
- MUX1
- MUX2
- c1
- c2

Table 2
- type
- cN

Table N
- type
- cN

Information vector

Index 1
- pc
- weights
- Sum1

Index 2
- LHR
- weights
- Sum2

Index N
- GHR -> PC
- weights
- SumN

Workload Detector
- type

Adaptation Logic
- (c1, c2, c3, ...)

Large Sum1 => strong correlation with PC bits
Small SumN => weak correlation with certain GHR bits
cN -> ‘PC’ => more PC bits
Overall scheme

- Detector
- Loop hit
- Info.
- Workload
- Predictor
- MAC perceptron
- LHR table
- GHR
- Loop branch predictor
- Bias branch predictor
- Initial (predict NT)
- Not biased (use other predictors)
- always Taken (predict T)
- always NT (predict NT)
- NT
- Taken
- Prediction
- loop prediction
Summary

• Observations
  – Different workloads/branches need different information.
  – Perceptron weights ↔ Correlation

• Contributions
  – Profile-directed adaptation
  – Correlation-directed adaptation
  – Reducing aliasing from bias and loop branches

• Result
  – Significant improvement
Thank you and Questions?
References


# Predictor configuration

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>CONFIGURATION</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias branch predictor</td>
<td>2293 entries</td>
<td>$2293 \times 2 = 4586$ bits.</td>
</tr>
<tr>
<td>Loop branch predictor</td>
<td>24 entries, 8-way set associative</td>
<td>1344 bits</td>
</tr>
<tr>
<td>Information</td>
<td>PC</td>
<td>$32 + 100 + 8 \times 63 = 636$ bits</td>
</tr>
<tr>
<td></td>
<td>GHR: 100 bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LHR: 8 bits, 63 entries</td>
<td></td>
</tr>
<tr>
<td>MAC perceptron predictor</td>
<td><strong>W0 table</strong>: 61 entries, 8 bits each</td>
<td>$61 \times 8 +$</td>
</tr>
<tr>
<td></td>
<td>Other table sizes: 63, 55, 53, 51, 49, 43, 41, 41, 39, 37, 37, and 35.</td>
<td>$597 \times 16 \times 6 +$</td>
</tr>
<tr>
<td></td>
<td><strong>Total MAC entries</strong>: 597</td>
<td>$597 \times 2$</td>
</tr>
<tr>
<td></td>
<td>Each entry: 16 weights, 6 bits each</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Control bits</strong>: 2 bits each entry</td>
<td>$= 58994$ bits</td>
</tr>
<tr>
<td>Adaptation</td>
<td>Adaptation interval: $100000 \times 2$ conditional branches</td>
<td>22 bits</td>
</tr>
<tr>
<td>Workload detection</td>
<td>Interval: $10000$ (instructions / conditional branches)</td>
<td>82 bits</td>
</tr>
<tr>
<td></td>
<td>Total: $65649$ bits (less than $64 \times 1024 + 256 = 65792$ bits)</td>
<td></td>
</tr>
</tbody>
</table>