Wormhole branch prediction using multi-dimensional histories

Jorge Albericio,
Joshua San Miguel,
Natalie Enright Jerger, and Andreas Moshovos
Motivation

```c
foreach frame:
    foreach object:
        if distance(object, p) < threshold:
            { /* do something */ }
```
Motivation

```plaintext
foreach frame:
    foreach object:
        if distance(object, p) < threshold:
            { /* do something */ }
```
Motivation

\[
\text{foreach frame:} \\
\quad \text{foreach object:} \\
\quad \quad \text{if distance(object, p) < threshold:} \\
\quad \quad \quad \{ /* do something */ \}
\]
Motivation

\textbf{foreach} frame:
\textbf{foreach} object:
\textbf{if} distance(\textbf{object}, \textbf{p}) < \textbf{threshold}:
\{ /* do something */ \}
Multidimensional histories

```plaintext
foreach frame:
  foreach object:
    if distance(object, p) < threshold:
      { /* do something */ }
```
Multidimensional histories

```plaintext
foreach frame:
  foreach object:
    if distance(object, p) < threshold:
      { /* do something */ }
```

![Diagram showing multidimensional histories with objects and distances](image)
Multidimensional histories

```
foreach frame:
  foreach object:
    if distance(object, p) < threshold:
      /* do something */
```
Multidimensional histories

foreach frame:
  foreach object:
    if distance(object, p) < threshold:
      /* do something */
Multidimensional histories

```plaintext
foreach frame:
  foreach object:
    if distance(object, p) < threshold:
      /* do something */
```

### Diagram

Frame 0
```
Object 0
1
Object 1
0
Object 2
1
Object 3
0
Object 4
1
```

Frame 1
```
Object 0
1
Object 1
0
Object 2
0
Object 3
1
Object 4
0
```

Frame 2
```
Object 0
1
Object 1
0
Object 2
1
Object 3
0
```

Object 4
```
Multidimensional histories

```
foreach frame:
    foreach object:
        if distance(object, p) < threshold:
            {/* do something */}
```
Multidimensional histories

foreach frame:
  foreach object:
    if distance(object, p) < threshold:
      /* do something */
Multidimensional histories

foreach frame:
    foreach object:
        if distance(object, p) < threshold:
            /* do something */
Summary

• **Wormhole predictor**: Side-predictor on top of ISL-TAGE

• Able to capture columns, diagonals, and other patterns

• **3.995, 2.498, and 2.014** MPKI in the 4KB, 32KB, and unlimited competition tracks
Detecting diagonals

```python
for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
```
Detecting diagonals

```python
for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
```
Detecting diagonals

```python
for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
```

```
1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
```
for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
Detecting diagonals

for j in N:  
    //Loop 1
    for k in N:  //Loop 2
        if j != k:
            /*Calculation*/
Detecting diagonals

Outer loop (j)
Iterations

Inner loop (k)
Iterations

\[
\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
\end{array}
\]

\textbf{for } j \text{ in } N: \quad // \text{Loop 1}
\textbf{for } k \text{ in } N: \quad // \text{Loop 2}
\textbf{if } j != k:\
\quad /* \text{Calculation}*/
Multidimensional histories

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
Multidimensional histories

for j in N:
  for k in N:
    if j != k:
      /*Calculation*/
Multidimensional histories

Outer loop (j) Iterations

Inner loop (k) Iterations

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
Multidimensional histories

Outer loop (j) Iterations

Inner loop (k) Iterations

Saturating counters

```python
for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
```
Multidimensional histories

```python
for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
```
Multidimensional histories

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
Multidimensional histories

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
Multidimensional histories

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/

Outer loop (j) Iterations

Inner loop (k) Iterations

Saturating counters
Multidimensional histories

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
Multidimensional histories

Outer loop (j) Iterations

Inner loop (k) Iterations

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/

Saturating counters

1
2
3
Multidimensional histories

```python
for j in N:
    for k in N:
        if j != k:
            /*Calculation*/
```
Multidimensional histories

\[
\text{for } j \text{ in } N:\n\text{for } k \text{ in } N:\n\text{if } j \neq k:\n\text{/*Calculation*/}
\]
Multidimensional histories

for j in N:
  for k in N:
    if j != k:
      /*Calculation*/
Multidimensional histories

for j in N:
    for k in N:
        if j != k:
            /*Calculation*/

for j in N:
    /*Calculation*/
Wormhole predictor and ISL-TAGE

- On top of ISL-TAGE (Best performer from last CBP)
  - TAGE: bimodal + tagged
  - Statistical corrector
  - Loop predictor
Wormhole predictor and ISL-TAGE

- On top of ISL-TAGE (Best performer from last CBP)
  - TAGE: bimodal + tagged
  - Statistical corrector
  - Loop predictor

Detect difficult branches
Wormhole predictor and ISL-TAGE

- On top of ISL-TAGE (Best performer from last CBP)
  - TAGE: bimodal + tagged
  - Statistical corrector
  - Loop predictor

Detect difficult branches

Loop dimensionality
ISL-TAGE

- **Difficult branches**
  - Statistical corrector calculates accuracy for TAGE
  - Wormhole allocates when low accuracy

- **Loop dimensionality**
  - Stored at $LP_{iters}$

- **$u$ field (4KB)**
  - 3 bits
  - Updated when “as in the original TAGE”
  - Not reset: Incremented and decremented
Wormhole predictor’s configurations

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Wormhole predictor’s configurations

```
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 0
```

4KB
Wormhole predictor’s configurations

4KB

32KB & Unlim.
Wormhole predictor’s configurations

4KB

32KB & Unlim.

PC Conf

Local history bits

Saturating counters
Wormhole predictor’s configurations

<table>
<thead>
<tr>
<th>4KB:</th>
<th>18</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>32KB &amp; Unlim.:</td>
<td>18</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>32KB &amp; Unlim.:</th>
<th>16 x 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(bits)</td>
<td></td>
</tr>
</tbody>
</table>

```
1     0     0     0     0     0     0
0     1     0     0     0     0     0
1     0     0     1     0     0     0
0     0     0     ?     0     0     0
```

```
1     0     0     0     0     0     0
0     1     0     0     0     0     0
1     0     0     1     0     0     0
0     0     0     ?     0     0     0
```
Wormhole predictor’s configurations

4KB: 18 4 101
32KB & Unlim.: 18 4 301

16 x 5 = 197 (bits)
256 x 5 = 1615 (bits)
Results: 32KB
Results: 32KB
Results: 32KB

19%

42%
Results: 32KB
Results: 32KB

Big improvements on two traces

32KB: Avg. MPKI = 2.498
Results: 4KB
Results: 4KB
Results: 4KB
Results: 4KB

Better than ISL-TAGE in 20 out of 40 traces (4.6% on avg.)

4KB: Avg. MPKI = 3.995
Wormhole predictor’s storage

<table>
<thead>
<tr>
<th></th>
<th>4KB</th>
<th>32KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical corrector</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>Loop predictor</td>
<td>376</td>
<td>376</td>
</tr>
<tr>
<td>TAGE predictor</td>
<td>3589</td>
<td>30476</td>
</tr>
<tr>
<td>Wormhole predictor</td>
<td>134</td>
<td>1413</td>
</tr>
<tr>
<td><strong>Total size (bytes)</strong></td>
<td><strong>4195</strong></td>
<td><strong>32649</strong></td>
</tr>
</tbody>
</table>
Conclusion

- **Wormhole predictor**: Side-predictor on top of ISL-TAGE

- Able to consider the branch history as multidimensional, capturing columns, diagonals and other patterns

- **3.995, 2.498, and 2.014** MPKI in the 4KB, 32KB, and unlimited competition tracks, respectively
Conclusion

• **Wormhole predictor**: Side-predictor on top of ISL-TAGE

• Able to consider the branch history as multidimensional, capturing columns, diagonals and other patterns

```
1 0 1 0 0
1 0 1 0 0
1 0 1 0 0
```

• **3.995, 2.498, and 2.014** MPKI in the 4KB, 32KB, and unlimited competition tracks, respectively
Wormhole branch prediction using multi-dimensional histories

Jorge Albericio, Joshua San Miguel, Natalie Enright Jerger, and Andreas Moshovos

Questions?