Measuring Memory Hierarchy Effects by Region

Bryan Buck
Jeffrey K. Hollingsworth

© Copyright 1998, Jeffrey K. Hollingsworth, All Rights Reserved.
Measuring Cache Effects by Region

- Simple base/bound register
  - Duplicate cache related performance counters
  - Each counter set collects info in own base/bounds
  - Difficult to convince chip makers to include

(assumes counters can use virtual, not physical addresses)
How to Measure

● **Use a software cache simulator**
  – Instrument applications to keep statistics
  – We’ve experimented with this using ATOM

● **Use cache miss counters with location info**
  – Keep track of regions in software (MIPS R10000)

● **Wait for processors with region counters**
  – MIPS R12000 (no OS support yet)
  – Intel Merced
Memory Hotspot Search

- Goal: identify region causing most misses
- Use $n$-way search
  - Start with all memory split $n$ ways and narrow down
  - Sample counters at regular intervals and readjust
  - Question: how does $n$ affect the results?

- Tested on SPEC95 benchmark applications
<table>
<thead>
<tr>
<th>application</th>
<th>variable</th>
<th>% of misses</th>
<th>2-way</th>
<th>10-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>tomcatv</td>
<td>RX</td>
<td>23.59</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>RY</td>
<td>23.57</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>DD</td>
<td>9.71</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>9.60</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>7.72</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UOLD</td>
<td>7.70</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VOLD</td>
<td>7.70</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UNEW</td>
<td>7.69</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PNEW</td>
<td>7.69</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POLD</td>
<td>7.69</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>7.64</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
## Search Results Continued

<table>
<thead>
<tr>
<th>application</th>
<th>variable</th>
<th>% of misses</th>
<th>2-way</th>
<th>10-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>su2cor</td>
<td>U</td>
<td>43.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W1</td>
<td>11.47</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>9.82</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>W2</td>
<td>9.04</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>mgrid</td>
<td>U</td>
<td>41.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>40.88</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>applu</td>
<td>B</td>
<td>21.23</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>21.22</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>21.22</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>compress</td>
<td>htab</td>
<td>66.49</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>codetab</td>
<td>25.81</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>jpeg</td>
<td>jpeg_com...</td>
<td>3.92</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Arrays Crossing Search Boundaries

- An array may span two or more regions
  - Not enough misses in single region for detection
  - This is the problem with su2cor
Search Time

- tomcatv
- swim
- su2cor
- mgrid
- applu
- compress
- ijpeg

Billion cycles

2-way
10-way
Misses vs. Time: Applu

sample interval (25 million cycles each)

A, B, C
U
RSD

University of Maryland
Results of Early Experiments

- Region miss information is useful
  - Automatic search can efficiently find arrays
- Phases are a problem
- More counters are more useful
  - 10-way search gets better results than 2-way
  - More counters doesn’t mean faster solution
- Cost of software instrumentation is high
  - Due to executing cache simulator every load/store
  - Much less instrumentation needed with hardware
Future Work

- Port search to Dyninst API
- Use hardware counters
  - MIPS R10000
- More sophisticated algorithms
  - Deal with phases
  - Better handling of dynamically allocated memory
    - Rearrange allocation for measurement
Misses vs. Time: Compress

- codetab
- comp_text_buffer
- orig_text_buffer
- htab

Sample interval (25 million cycles each)