**Problem Definition**

To propose a hardware prefetcher for parallel applications at L1 D-cache level for Chip Multiprocessors (CMPs).

**Motivation**

In parallel applications, multiple threads share and communicate data with each other. Per-core data prefetchers are oblivious to access patterns observed at other cores. Local prefetchers are unaware of irregular spread of misses across the cores.

**Issues in Existing Prefetchers**

- Low prefetch issue rate.
- Predicted addresses cross the virtual page boundary.
- Per core local strides are less uniform.
- Length of data streams are short and non-uniform.

**What is desired?**

They do well within a core

<table>
<thead>
<tr>
<th>Thread Id</th>
<th>Miss Address</th>
<th>Stride</th>
<th>Phase</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>-</td>
<td>Training</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>100</td>
<td>Training</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>100</td>
<td>Training</td>
<td>1</td>
</tr>
</tbody>
</table>

Not so well across cores: Lost opportunities

<table>
<thead>
<tr>
<th>Thread Id</th>
<th>Miss Address</th>
<th>Local Stride</th>
<th>Global Stride</th>
<th>Phase</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>Training</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>100</td>
<td>-</td>
<td>Training</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>-</td>
<td>100</td>
<td>Training</td>
<td>0</td>
</tr>
</tbody>
</table>

**Our Classification of Misses**

- Threads $i$ and $j$; line $L_i$; stride $S$
- Global Stride Miss (GS):
  \[ (L'_i)(L \pm S)_{j \neq i} \]
- Local Stride (LS):
  \[ (L'_i)(L \pm S)_{j \neq i} \]
- Producer Consumer Shared Miss (PCS):
  \[ (L'_i)(L)_{j \neq i} \]

**Spread of Non-Local Strides Across Threads**

On an average, more than 50% misses spread across threads

**Prefetcher Framework**

**GS and PCS**

- GS and PCS prefetchers are shared by all the threads.
- Both hide miss latency (not miss rate)

**Sequence Diagram of Events on a Miss**

**Results**

- gem5 FS simulator, PARSEC Benchmarks, Sim medium inputs, ROI
- 32KB L1, 2/4 MB shared L2, MOESI, 8/16 MSHRs at L1/L2
- On an average, our prefetcher outperforms feedback based prefetcher [1] by 4%/7% for 2/(4) core systems.

**Reference**