Stream Chaining: Exploiting Multiple Levels of Correlation in Data Prefetching

Pedro Díaz and Marcelo Cintra

University of Edinburgh
http://www.homepages.inf.ed.ac.uk/mc/Projects/CELLULAR
Outline

- Motivation
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- Related Work
- Conclusions
The “Memory Wall” and Prefetching

- The Memory Wall is still a problem
  - After decades of logic and DRAM technology disparity, memory access costs hundreds of processor cycles
  - On-chip cache quotas per processor unlikely to increase
  - Off-chip memory bandwidth quota per processor likely to decrease (unless some fancy memory technology succeeds)

- (Hardware) Prefetching is a viable solution
  - Time-tested approach used in most commercial processors
  - Trades-off memory bandwidth for latency (especially good if some fancy memory technology succeeds)
Prefetching

- Prefetchers work by uncovering patterns in the miss address stream: *correlation* (e.g., address deltas)
- Prefetchers often separate misses into multiple streams: *localization* (e.g., by instruction)
- To eliminate more misses and hide longer latencies prefetchers often use prefetch degree greater than one
- Prefetchers often measured against three metrics:
  - Accuracy: ratio of used prefetches over all prefetches
  - Coverage: ratio of used prefetches over original misses
  - Timeliness: data arrives too early, too late, or just in time
The Problem with Prefetching

- Correlation on global miss stream often suffers from poor accuracy
- Prefetching along localized streams often suffers from poor coverage and timeliness
  - Streams lose time ordering information of misses
  - “Cold” misses across stream boundaries
- Deep prefetching suffers from diminishing accuracy
- Applications access patterns exhibit different correlation patterns

Ideally what we want is to combine multiple localized streams to improve coverage and timeliness while keeping accuracy high
Outline

- Motivation
- **Correlation and Localization**
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- Related Work
- Conclusions
Correlation

- Establishing “relationship” among addresses of misses. For instance:
  - Sequential: miss to line L is followed by miss to line L+1
  - Time: miss to address A is followed by miss to address B
  - Delta: miss to address A is followed by miss to address A +d
  - Markov: e.g., miss to address A is followed by miss to address B with probability p and miss to address C with probability (1-p)

- Correlations are found by inspecting miss history and are used to predict next miss
Localization

- Complete global history is undesirable in most cases
  - Misses from unrelated sources (e.g., from pointer chasing followed by data object manipulation)
  - “Wild” interleaving of misses (e.g., OOO execution, infrequent control flow)
  - Correlations over long traces

- Localization: group misses according to some common property. For instance:
  - PC: misses from same static instruction
  - Temporal: misses that occur at about the same time
  - Spatial: misses to similar regions in memory address space

- Attempts to exploit some high-level behaviour
Localization

Miss Stream (PC : Addr)

PC Correlation

PC Localized Streams:
A1 → A7 → A1 → A11 → A1 → A7
A2 → A8 → A2 → A12 → A2 → A8

Memory Address Space

A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14

ISCA 2009
Localization

Miss Stream (PC : Addr)

<table>
<thead>
<tr>
<th>PC_A</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC_B</td>
<td>A2</td>
</tr>
<tr>
<td>PC_A</td>
<td>A7</td>
</tr>
<tr>
<td>PC_D</td>
<td>A5</td>
</tr>
<tr>
<td>PC_B</td>
<td>A8</td>
</tr>
<tr>
<td>PC_A</td>
<td>A1</td>
</tr>
<tr>
<td>PC_B</td>
<td>A2</td>
</tr>
<tr>
<td>PC_C</td>
<td>A4</td>
</tr>
<tr>
<td>PC_E</td>
<td>A6</td>
</tr>
<tr>
<td>PC_A</td>
<td>A11</td>
</tr>
<tr>
<td>PC_B</td>
<td>A12</td>
</tr>
<tr>
<td>PC_A</td>
<td>A1</td>
</tr>
<tr>
<td>PC_B</td>
<td>A2</td>
</tr>
<tr>
<td>PC_A</td>
<td>A7</td>
</tr>
<tr>
<td>PC_B</td>
<td>A8</td>
</tr>
</tbody>
</table>

Temporal Correlation

Time Localized Streams:
A1 → A2 → A7 → A5 → A8
A1 → A2 → A4 → A6 → A11 → A12
A1 → A2 → A7 → A8

Memory Address Space

A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14

ISCA 2009
Localization

Miss Stream (PC : Addr)
- PC_A : A1
- PC_B : A2
- PC_A : A7
- PC_D : A5
- PC_B : A8
- PC_A : A1
- PC_B : A2
- PC_C : A4
- PC_E : A6
- PC_A : A11
- PC_B : A12
- PC_A : A1
- PC_B : A2
- PC_A : A7
- PC_B : A8

Spatial Correlation
- A1 → A2
- A1 → A2 → A4
- A7 → A8
- A11 → A12

Space Localized Streams:

Memory Address Space
- A1
- A2
- A3
- A4
- A5
- A6
- A7
- A8
- A9
- A10
- A11
- A12
- A13
- A14

ISCA 2009
Outline

- Motivation
- Correlation and Localization
- **Stream Chaining and Miss Graph Prefetching**
- Experimental Setup and Results
- Related Work
- Conclusions
Stream Chaining: Idea and Operation

- Chain streams:
  - Start from global, ordered, miss stream
  - Perform localization and build localized streams
  - Order and link streams according to program execution to partially reconstruct order of misses

- Prefetch
  - On a miss to stream A follow chain and identify streams that commonly follow A
  - Perform correlation on each stream individually
  - Prefetch data for streams that follow A and, possibly, also for A itself
Benefits and Limitations

+ Recover chronological information following program’s *stable* memory access pattern
+ Still eliminate “spurious” misses
+ Still benefit from better predictability of localized streams
+ Prefetch across stream boundaries
+ Better use of large prefetch degrees
  - Stream chain patterns must be stable
  - Stream chains must be relatively small as to be manageable
  - Longer run time of algorithm as must correlate on multiple streams
Miss Graph Prefetcher

- Based on Nesbitt and Smith’s GHB structure (HPCA’04)
- Uses PC localization with delta correlation (PC/DC)
- Represents stream chains as simple directed graphs
  - Nodes represent streams and edges represent time ordering (i.e., miss to stream A is followed by miss to stream B → A → B)
  - Only 1 outgoing edge per node but multiple incoming edges possible
  - Edges only added to recurring sequences by using a threshold
  - Cycles allowed

Named PC/DC/MG
Miss Graph Prefetcher

Miss Stream (PC : Addr)
- PC_A : A1
- PC_B : B1
- PC_C : C1
- PC_D : D1
- PC_E : E1
- PC_A : A2
- PC_D : D2
- PC_E : E2
- PC_A : A3
- PC_D : D3
- PC_E : E3
- PC_A : A4

Index Table

Global History Buffer

Global History Buffer

PC_A

PC_B

PC_C

PC_D

PC_E

PC_A

PC_B

PC_C

PC_D

PC_E

PC_A

PC_B

PC_C

PC_D

PC_E

A 1

B 1

C 1

D 1

E 1

A 2

A 3

D 2

D 3

E 2

E 3

E 4

PC_A

PC_B

PC_C

PC_D

PC_E

ISCA 2009
Miss Graph Prefetcher

- Step 1: perform localization → already part of GHB funct.
Miss Graph Prefetcher

- **Step 2: chain streams**

```plaintext
Miss Stream (PC : Addr)
PC_A : A1
PC_B : B1
PC_C : C1
PC_D : D1
PC_E : E1
PC_A : A2
PC_D : D2
PC_E : E2
PC_A : A3
PC_D : D3
PC_E : E3
PC_A : A4

Index Table

<table>
<thead>
<tr>
<th>Miss Stream (PC : Addr)</th>
<th>Next</th>
<th>Ctr</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC_A</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PC_B</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PC_C</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PC_D</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PC_E</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Global History Buffer

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>B1</th>
<th>C1</th>
<th>D1</th>
<th>E1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global History Buffer</td>
<td>A2</td>
<td>D2</td>
<td>E2</td>
<td>A3</td>
<td>D3</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>A4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- time
- current miss

ISCA 2009
Miss Graph Prefetcher

- **Step 2: chain streams**

<table>
<thead>
<tr>
<th>Miss Stream (PC : Addr)</th>
<th>Index Table</th>
<th>Global History Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC_A : A1</td>
<td>PC_A</td>
<td>A 1</td>
</tr>
<tr>
<td>PC_B : B1</td>
<td>PC_B</td>
<td>B 1</td>
</tr>
<tr>
<td>PC_C : C1</td>
<td>PC_C</td>
<td>C 1</td>
</tr>
<tr>
<td>PC_D : D1</td>
<td>PC_D</td>
<td>D 1</td>
</tr>
<tr>
<td>PC_E : E1</td>
<td>PC_E</td>
<td>E 1</td>
</tr>
<tr>
<td>PC_A : A2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC_D : D2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC_E : E2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC_A : A3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC_D : D3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC_E : E3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC_A : A4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Index Table:
- Current miss: PC_A
- Next PC: PC_B
- Ctr: 1

Global History Buffer:
- A 1
- B 1
- C 1
- D 1
- E 1
- A 2
- D 2
- E 2
- A 3
- D 3
- E 3
- A 4
Miss Graph Prefetcher

- Step 2: chain streams
Miss Graph Prefetcher

- Step 2: chain streams

Miss Stream (PC : Addr)
- PC_A : A1
- PC_B : B1
- PC_C : C1
- PC_D : D1
- PC_E : E1
- PC_A : A2
- PC_D : D2
- PC_E : E2
- PC_A : A3
- PC_D : D3
- PC_E : E3
- PC_A : A4

Index Table
- PC_A
- PC_B
- PC_C
- PC_D
- PC_E

Next
- 1

Ctr
- 1

Global History Buffer
- A 1
- B 1
- C 1
- D 1
- E 1
- A 2
- D 2
- E 2
- A 3
- D 3
- E 3
- A 4

ISCA 2009
Miss Graph Prefetcher

- Step 2: chain streams

Miss Stream (PC : Addr)

| PC_A : A1 | PC_B : B1 | PC_C : C1 |
| PC_D : D1 | PC_E : E1 |
| PC_A : A2 | PC_D : D2 | PC_E : E2 |
| PC_A : A3 | PC_D : D3 | PC_E : E3 |
| PC_A : A4 |

time

current miss

Index Table

<table>
<thead>
<tr>
<th>PC_A</th>
<th>PC_B</th>
<th>PC_C</th>
<th>PC_D</th>
<th>PC_E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Global History Buffer

<table>
<thead>
<tr>
<th></th>
<th>A 1</th>
<th>B 1</th>
<th>C 1</th>
<th>D 1</th>
<th>E 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ctr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PC_A
PC_B
PC_C
PC_D
PC_E
Miss Graph Prefetcher

- **Step 2: chain streams**

Miss Stream (PC : Addr)

| PC_A : A1 |
| PC_B : B1 |
| PC_C : C1 |
| PC_D : D1 |
| PC_E : E1 |
| PC_A : A2 |
| PC_B : B2 |
| PC_C : C2 |
| PC_D : D2 |
| PC_E : E2 |
| PC_A : A3 |
| PC_B : B3 |
| PC_C : C3 |
| PC_D : D3 |
| PC_E : E3 |
| PC_A : A4 |

Index Table

<table>
<thead>
<tr>
<th>PC_A</th>
<th>PC_B</th>
<th>PC_C</th>
<th>PC_D</th>
<th>PC_E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

Global History Buffer

<table>
<thead>
<tr>
<th>PC_A</th>
<th>PC_B</th>
<th>PC_C</th>
<th>PC_D</th>
<th>PC_E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
</tbody>
</table>

**current miss**

**time**

ISCA 2009
Miss Graph Prefetcher

- Step 3: perform correlations and prefetch along streams

Note that we do not prefetch for A, but rely on “peers” (i.e., D and/or E) to prefetch for A
Miss Graph Example

- perlbench (512KB L2)
Outline

- Motivation
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- **Experimental Setup and Results**
- Related Work
- Conclusions
Experimental Setup

- **Simulator:**
  - SESC: cycle-accurate architectural simulator from UIUC

- **Applications:** SPEC2006 and BioBench

- **Architecture:**
  - 5GHz, 4-issue superscalar MIPS processor
  - 64KB, 2-way L1 I-Cache and 64KB, 2-way L1 D-Cache
  - 256KB/2MB, 8-way L2 cache
  - 64bit, 1.25GHz memory bus
  - Main memory: 400 cycle latency
Performance Without Prefetching

Many applications still perform poorly
Some perform well

Some even with large (2MB) L2
Others even with 512KB L2

Some applications already perform well
within 15% of ideal
Performance With Prefetching

Best performing prefetching scheme varies across applications. Overall, PC/DC/MG performs best or close to best in most applications.
Prefetch Coverage

PC/DC often has lowest coverage, and PC/DC/MG and G/DC vary across applications
... and Accuracy

PC/DC/MG is often the most accurate, and PC/DC is often more accurate than G/DC
## Miss Graphs Statistics

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Unique Subgraphs (%)</th>
<th>Nodes</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Snapshot</td>
<td>CC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>max</td>
<td>avg.</td>
<td>max</td>
</tr>
<tr>
<td>milc</td>
<td>4.7</td>
<td>15</td>
<td>7.7</td>
<td>7</td>
</tr>
<tr>
<td>lbm</td>
<td>22</td>
<td>20</td>
<td>7.9</td>
<td>18</td>
</tr>
<tr>
<td>lbq</td>
<td>0.8</td>
<td>23</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>zeusmp</td>
<td>11</td>
<td>18</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>clustalw</td>
<td>1.1</td>
<td>10</td>
<td>9.3</td>
<td>10</td>
</tr>
<tr>
<td>perl</td>
<td>11</td>
<td>16</td>
<td>8.6</td>
<td>9</td>
</tr>
<tr>
<td>namd</td>
<td>21</td>
<td>8</td>
<td>5.8</td>
<td>8</td>
</tr>
<tr>
<td>soplex</td>
<td>2.8</td>
<td>30</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>bzip2</td>
<td>5.6</td>
<td>38</td>
<td>20</td>
<td>9</td>
</tr>
<tr>
<td>tiger</td>
<td>5.4</td>
<td>41</td>
<td>30</td>
<td>18</td>
</tr>
<tr>
<td>hmmer</td>
<td>12</td>
<td>50</td>
<td>38</td>
<td>33</td>
</tr>
<tr>
<td>gobmk</td>
<td>20</td>
<td>10</td>
<td>5.2</td>
<td>5</td>
</tr>
</tbody>
</table>

Most graphs appear repeatedly during execution, thus offering potential for learning.

Moreover, the graphs are stable for long periods of time, offering potential to exploit patterns.

The number of nodes at any given time is small, making it manageable to keep track of.

Number of nodes per stream groups is small, leading to small protocol execution overheads.

---

ISCA 2009
Next-Stream Prediction Accuracy

Miss-graph’s prediction accuracy is often very high
Outline

- Motivation
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- Related Work
- Conclusions
(Closest) Related Work

- K. Nesbit and J. Smith – HPCA’04
  - Proposed GHB and introduced PC/DC

- S. Somogyi, T. Wenisch, A. Ailamaki, and B. Falsafi – ISCA’09
  - Combined spatial and temporal memory streaming
  - Can be seen as close to a PID/SMS/TMS prefetcher (except that PID is not used to index at prefetch time)
Outline

- Motivation
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- Related Work
- Conclusions
Conclusions

- New strategy for creating prefetchers by composing (chaining) localization and correlation schemes
- New prefetcher based on the Stream Chaining idea
  - Simple extension of GHB-based PC/DC of Nesbit and Smith (HPCA’04)
  - Captures most of the stable miss sequences in the programs tested
  - Overall better performance than PC/DC or G/DC
- Stream Chaining could be applied to other localization and correlation schemes (we are working on it)
Stream Chaining: Exploiting Multiple Levels of Correlation in Data Prefetching

Pedro Díaz and Marcelo Cintra

University of Edinburgh
http://www.homepages.inf.ed.ac.uk/mc/Projects/CELLULAR
Miss Distances

Global miss distances are often in the order of tens or hundreds of cycles only.

PC localized miss distances are always much larger, often in the order of thousands or tens of thousands.
Miss graph prefetching

- **Prefetch operation**

  Not long enough or cyclic chains: Prefetch degree/length items per stream item from PC_B onwards.

  - Long enough linear stream: Prefetch 1 item from PC_B onwards.

  - (a)
    - PC_A
    - PC_B
    - PC_H
    - PC_I
  
  - (b)
    - PC_A
    - PC_B
    - PC_C
    - PC_D
  
  - (c)
    - PC_A
    - PC_B
    - PC_C
    - PC_D
Miss Graph examples

- bzip2 (2048KB L2)
Miss Graph examples

- lbm (512KB L2)
Miss Graph examples

- libquantum (256KB L2)