Spatio-Temporal Memory Streaming

Stephen Somogyi, Thomas F. Wenisch, Anastasia Ailamaki and Babak Falsafi

36th International Symposium on Computer Architecture
June 22, 2009
Memory Wall: Performance Bottleneck

Especially for server apps
• Large data footprints
• Pointer-intensive structures

Prefetching can hide long memory latency
[ISCA ’05] [ISCA ’06] [Micro ’07]

But, no single technique effective for OLTP, DSS & Web
Observation: Temporal, Spatial Predictions Different

Different predictors capture different behaviors

- Temporal: recurring memory access sequences (pointers)
- Spatial: recurring data layouts (structs)

Spatial/temporal disjoint; opportunity to exploit both
How to Combine?

**Concept:** independent temporal & spatial
- Duplicates prefetches $\Rightarrow$ inefficient
- Prefetchers interfere $\Rightarrow$ confuses training

**Refinement:** chain spatial predictions via temporal
- No sequence within spatial predictions
- Prefetches in wrong order $\Rightarrow$ not timely

**Solution:** also learn order within spatial predictions
- Achieves high, consistent coverage & low mispredictions
Contributions

• Opportunity for spatio-temporal prediction
  – 70% of read misses are predictable on average

• Temporal characterization of spatial accesses
  – Sequences repetitive within & across layouts

• Spatio-Temporal Memory Streaming
  – Predicts unified spatio-temporal miss sequence
    • 62% of read misses on average
  – Mean speedup 1.31, ≥ temporal or spatial alone
Outline

• Introduction

• Spatial and Temporal Prediction

• Spatio-Temporal Memory Streaming

• Results

• Conclusion
Example: Non-Clustered Index Scan

Sequence spans non-contiguous data pages
Similar layouts within data pages
Temporal Memory Streaming (TMS)
[Wenisch ’05]

Records & replays recurring miss sequences
• Code traversals repeat ⇒ data traversals repeat

Sequences contain entire addresses
✓ Good for pointer chasing ⇒ breaks dependence chains
✓ Startup costs amortized over long streams
✗ Cannot predict compulsory misses
✗ Large storage required (~2MB / processor)
Spatial Memory Streaming (SMS)  
[Somogyi '06]  

Exploits repetitive, large-scale data layouts in memory  
- Pattern: offsets in logical region  
- Trigger: first miss, used for lookup  

Patterns encoded as bit vectors  
- PC lookup ⇒ predicts compulsory misses  
- Efficient storage (~80KB / processor)  
- Trigger miss per pattern ⇒ lost opportunity  
- Unordered ⇒ BW spikes, wrong prioritization
Hybrid Spatio-Temporal Predictor

Naïve approach
  – Record sequence across patterns
  – Fetch entire pattern at a time
    • No priority across patterns

But, triggers many patterns simultaneously
  – Accesses across patterns are interleaved!
  – Bursts of prefetches ⇒ pollution, BW spikes

Must prioritize prefetches across patterns
Deconstructing a Miss Sequence

Investigate temporal & spatial relationships

overall miss sequence

<table>
<thead>
<tr>
<th>base addr + offset</th>
<th>A+7</th>
<th>A+4</th>
<th>B+5</th>
<th>A+2</th>
<th>B+6</th>
<th>A+0</th>
<th>C+3</th>
<th>D+3</th>
<th>E+0</th>
<th>E+2</th>
<th>D+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>skip = 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>skip = 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>skip = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>skip = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

sequence across patterns

<table>
<thead>
<tr>
<th>base addr + offset, offset</th>
<th>A+7,0</th>
<th>B+5,1</th>
<th>C+3,3</th>
<th>D+3,0</th>
<th>E+0,1</th>
</tr>
</thead>
</table>

sequences within patterns

<table>
<thead>
<tr>
<th>base addr</th>
<th>sequence: (offset) skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(4),0,2,2,0,1</td>
</tr>
<tr>
<td>B</td>
<td>(6),1</td>
</tr>
<tr>
<td>D</td>
<td>(1),2</td>
</tr>
<tr>
<td>E</td>
<td>(2),0</td>
</tr>
</tbody>
</table>

Naïve approach: prefetches pattern A before B
Incorrect order!

Reverse process to reconstruct the miss sequence
Spatio-Temporal Memory Streaming (STeMS)

**Goal:** reconstruct the overall miss sequence
- Using both temporal & spatial predictions
  ⇒ Prefetch cache blocks in order

**Training:** observe relative interleavings
- Record skips in temporal seq. and spatial patterns

**Prediction:** reconstruction buffer for staging
- Spread temporal predictions according to skips
- Trigger spatial lookup for each, insert predicted addresses

Generates simple address seq. for throttled streaming
Outline

• Introduction

• Spatial and Temporal Prediction

• Spatio-Temporal Memory Streaming

• Results

• Conclusion
Methodology

Flexus [Wenisch ‘06]
• Full-system trace and OoO timing simulation
• Leverages SMARTS sampling

Benchmark Applications
• OLTP: TPC-C
  – IBM DB2 & Oracle
• DSS: TPC-H Qry 2,16,17
  – IBM DB2
• Web: SPECweb99
  – Apache & Zeus
• Scientific
  – em3d, ocean, sparse

Model Parameters
- 16 4GHz SPARC CPUs
- 4-wide OoO; 96-entry ROB
- 64KB 2-way L1
- 8MB 8-way L2, 25-cycle lat.
- 40ns memory
- 25ns per-hop network
- TSO w/ speculation
Temporal Repetition Within Patterns

Compare access sequence of successive patterns
– Evaluate for small reordering windows

Seq. of accesses within patterns extremely repetitive
Temporal Repetition Across Patterns

Evaluate repetition with compression algorithm

- Past work (TMS): all miss addresses
- STeMS: only trigger misses

Similar opportunity for predicting seq. of triggers
STeMS Results

STeMS is effective across commercial workloads
Predicts 62% of misses, improves perf. 31%

OLTP/DSS – matches TMS/SMS, Web – beats both
Related Work

- **Stream Chaining** [Diaz ‘09]
  - Reconstruct overall miss seq. using control flow
  - Better compulsory, worse temporal coverage

- **Predictor Virtualization** [Burcea ‘08]
  - Reduces dedicated on-chip storage for predictors
  - Can be applied to history structures in STeMS

- **Epoch-base Correlation Prefetching** [Chou ‘07]
  - Improves timeliness of temporal predictions
  - In contrast, STeMS mainly targets spatial timeliness
Conclusion

• Spatial/temporal predictions disjoint
  – Opportunity to predict up to 70% of read misses

• Temporal repetition of spatial patterns
  – Near-perfect repetition within patterns
  – Similar repetition across patterns as all addresses

• Design for Spatio-Temporal Memory Streaming
  – Reconstructs total miss sequence
    • Predicts 62% of read misses, perf. improvement 31%
  – Coverage & speedup ≥ temporal or spatial alone
Questions?

STeMS Project
Spatio-Temporal Memory Streaming
www.ece.cmu.edu/~stems

Computer Architecture Laboratory
Carnegie Mellon University
www.ece.cmu.edu/~calcm