Matryoshka: A Coalesced Delta Sequence Prefetcher

Shizhi Jiang
Yiwei Ci
Qiusong Yang
Mingshu Li
Presentation Structure

- Background
- Motivation
- Design & Implementation
- Evaluation
- Conclusion
Background

The “Memory Wall”

Prefetcher
- learning access history
- predict future accesses
- low-overhead
- high-performance
Background

Delta Sequence

Delta = Current Addr - Last Addr

e.g. delta seq : 10, 5, 15

Recursive Lookahead

Advantages

a) Shared Among Different Regions
b) Record Access Order
Motivation

Why we need multi-matching?

➢ Ideal Coverage
  ➢ Always scattered in workloads

➢ Average Branch Number (Ideal Accuracy)
  ➢ Good in 3-delta or longer sequences

We can hardly decide the best delta sequence length!
Motivation

Better Form Storing Patterns

It's obvious more efficient if we can find a method to coalesce variable-length sequences
Motivation

Storing Hot Patterns

Only a small part of deltas reappear frequently
**Design**

Reversed Coalesced Delta Sequences

The confidence (repeating times) of a sequence is vital for accurate prefetch! How to deal with them during coalescing?

- With same confidences
  - Discard confidences of short sequences
- With different confidences

---

1. Store 3 sequences separately
2. A Coalesced Delta Sequence
3. Layout Before Reversing
4. Layout After Reversing
5. Coalesce + Reverse
Design

Dynamic Indexing Strategy & Overview

Dynamically record the mapping relations between deltas and table entries
Design

Adaptive Voting Strategy

\[
S\text{core}_d = \sum_{i \in L} W_i \sum_{f \in M_i} \text{Conf}_i 
\]

\[
\frac{\text{Score}_d}{\text{Score}_{total}} > T_p
\]

In our configuration,

\( T_p = 0.5 \) for prefetching into L1

\( W_2 = 3 \) and \( W_3 = 4 \) for 2-delta and 3-delta matched prefixes.

\( W_i \) is the weight for each matched prefix of length \( i \);

\( L \) is the set of the lengths;

\( M_i \) indicates the entries, containing the matched prefixes of length \( i \), that generate \( d \);

\( \text{Conf}_j \) is the confidence of the corresponding entry \( j \);

\( D \) is the set of candidate deltas.

Consider both long and short matches
Implementation

Training

History Table

<table>
<thead>
<tr>
<th>PC tag</th>
<th>Page tag</th>
<th>Last offset</th>
<th>Last delta sequence</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x9a540</td>
<td>0x37caabd0</td>
<td>80</td>
<td>20,22,24</td>
<td>1</td>
</tr>
</tbody>
</table>

Cur delta seq: 24,22,20,26

1. Cur offset=(0x37caabd0170>>3)&(2**9-1)=106
   Cur delta=106-last offset=106-80=26
   Cur delta seq= Reverse(Last delta seq) << 10 | cur delta

Pattern Table

<table>
<thead>
<tr>
<th>Way</th>
<th>Delta</th>
<th>Conf</th>
<th>Valid</th>
<th>Way</th>
<th>Delta Seq</th>
<th>Conf</th>
<th>Valid</th>
<th>Way</th>
<th>Delta Seq</th>
<th>Conf</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Way 0</td>
<td>22</td>
<td>4</td>
<td>1</td>
<td>Way 0</td>
<td>18.8,36</td>
<td>3</td>
<td>1</td>
<td>Way 0</td>
<td>25.20,36</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Way 1</td>
<td>16</td>
<td>5</td>
<td>1</td>
<td>Way 1</td>
<td>20.18,22</td>
<td>2</td>
<td>1</td>
<td>Way 1</td>
<td>20.18,24</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Way 2</td>
<td>24</td>
<td>7-8</td>
<td>1</td>
<td>Way 2</td>
<td>22.20,26</td>
<td>7-8</td>
<td>1</td>
<td>Way 2</td>
<td>0.0,0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Way 3</td>
<td>26</td>
<td>8</td>
<td>1</td>
<td>Way 3</td>
<td>24,22,28</td>
<td>8</td>
<td>1</td>
<td>Way 3</td>
<td>24,19,24</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Following sequence: 22,20,26

L1 load access:
PC:0x9a540
Addr:0x37caabd0170
## Implementation

### Predicting

#### DMA

<table>
<thead>
<tr>
<th>Way</th>
<th>Delta</th>
<th>Conf</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>22</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

#### DSS

<table>
<thead>
<tr>
<th>Way</th>
<th>Delta Seq</th>
<th>Conf</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>18,8,36</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>20,18,22</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>22,20,26</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>24,22,28</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Way</th>
<th>Delta Seq</th>
<th>Conf</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25,20,36</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>20,18,24</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0,0,0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>24,19,24</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Reversed current seq

Best delta: 28
Max Score: 8X4=32, 32/(8X4+3X3)=32/41>0.5
Evaluation: Methodology

Simulator: ChampSim

Benchmark: 45 traces from SPEC 2017, CloudSuite

Table 1: Detailed Storage Overhead of Matryoshka

<table>
<thead>
<tr>
<th>Structure</th>
<th>Entry</th>
<th>Field</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>History Table</td>
<td>128 x 1</td>
<td>PC tag (12 bits)</td>
<td>7680 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page tag (8 bits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Last offset (9 bits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Last delta sequence (30 bits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid (1 bit)</td>
<td></td>
</tr>
<tr>
<td>Delta Mapping Array</td>
<td>1 x 16</td>
<td>Delta (10 bits)</td>
<td>272 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Confidence (6 bits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid (1 bit)</td>
<td></td>
</tr>
<tr>
<td>Delta Sequence</td>
<td>16 x 8</td>
<td>Delta sequence (30 bits)</td>
<td>5120 bits</td>
</tr>
<tr>
<td>Sub-table</td>
<td></td>
<td>Confidence (9 bits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid (1 bit)</td>
<td></td>
</tr>
<tr>
<td>Candidate Array</td>
<td>128 x 1</td>
<td>Score (10 bits)</td>
<td>1280 bits</td>
</tr>
<tr>
<td>Candidate Offset</td>
<td>32 x 1</td>
<td>Score (10 bits)</td>
<td>320 bits</td>
</tr>
</tbody>
</table>

Total: 14,672 bits ≈ 1.79KB

Table 3: Prefetcher Overhead

<table>
<thead>
<tr>
<th></th>
<th>VLDP</th>
<th>SPP+PPF</th>
<th>Pangloss</th>
<th>IPCP</th>
<th>Matryoshka</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td>48.34 KB</td>
<td>48.39 KB</td>
<td>45.25 KB</td>
<td>740 B</td>
<td>1.79 KB</td>
</tr>
</tbody>
</table>

we expand VLDP’s storage capacity to 48 KB
**Evaluation: Single-core**

Matryoshka yields the best geometric mean speedup of 53.1% over the non-prefetching system, an improvement of 6.5% over IPCP.

- Compared with the other prefetchers, Matryoshka achieves the highest average coverage (57.4%) at L1, which exceeds the second best IPCP by 6.0%.
- The average overprediction rate for Matryoshka is 20.6%, which is the lowest.
Evaluation: Multicore

In the 4-core system, Matryoshka offers a 32.2% speedup over the baseline.
Compared with the other four prefetchers, Matryoshka still obtains better performance in different configurations of the cache system.
Conclusion

• Coalesced Delta Sequence
  • Simpler Structure for Multiple Matching

• Adaptive Voting Strategy
  • Ensure both coverage and accuracy

• Dynamic Indexing Strategy
  • Keep key information