Footprint Modeling of Cache Associativity and Granularity

Hao Luo¹ (UR), Guoyang Chen² (NCSU), Fangzhou Liu (UR), Pengcheng Li¹ (UR), Chen Ding (UR), Xipeng Shen (NCSU)

University of Rochester, Rochester, NY
North Carolina State University, Raleigh, NC

¹. Now at Google Inc. This work was done when the student was a graduate student
². Now at Alibaba Group US Inc. This work was done when the student was a graduate student
Motivation

Data placement problem

3X performance difference
Motivation

PORPLE\textsuperscript{1} - Online Optimization of Data Placement

1. Guoyang Chen, Bo Wu, Dong Li and Xipeng Shen. 2014. PORPLE: An Extensible Optimizer for Portable Data Placement on GPU. In Proceedings of MICRO.
Motivation

- **Data Non-uniformity**
  - A program uses multiple arrays, which have different access frequency and locality.

- **Uneven Address Mapping**
  - Data addresses may be mapped unevenly to cache sets.
  - Cache can be reconfigured to different capacity
    - Which changes address mapping

- **Dual-grained Cache**
  - Data was fetched and managed in different granularity.

```c
for (i = 1; i < 1025; i++) {
    for (j = 1; j < 1025; j++) {
        B[i][j] = A[i][j]
            + A[i][j+1]
            + A[i][j-1]
            + A[i-1][j]
            + A[i+1][j];
    }
}
```
Agenda

• Footprint Metric

• Three footprint models

• Evaluation
Background

Footprint Metric

- **Trace** is a series of memory accesses (interleaved)

  Example: \[\text{a b c a c d}\]

- **Working Set Size** is the number of distinct elements within a window (fixed-length time interval)

- **Footprint** is the average number of distinct items accessed in windows of length “x”

\[
fp(x) = \frac{1}{N - x + 1} \sum_{t=x}^{N} \omega(t, x)
\]
Background

Footprint Metric

Time:

1 2 3 4 5 6

Trace:

a b c a c d

\[ \text{AVG}(3, 3, 2, 3) = 2.75 \]

- **Footprint** is the average number of distinct items accessed in windows of length “x”

<table>
<thead>
<tr>
<th>Window length</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>≥ 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Footprint</td>
<td>1</td>
<td>2</td>
<td>2.75</td>
<td>3</td>
<td>3.5</td>
<td>4</td>
</tr>
</tbody>
</table>
The **Cache Miss ratio** can be computed by the derivative of footprint function.

\[ \text{mr}(c) = \frac{d}{dx} fp(x) \bigg|_{fp(x) = c} \]

**Denning-HOTL Conversion**

\[ \text{mr}(c) = \text{slope} = fp(x+1) - fp(x) \bigg|_{fp(x) = c} \]

Agenda

• Footprint Metric

• Three new footprint models
  • Nonuniform data locality -> Partial Footprint (This talk)
  • Uneven address mapping -> Mapped Footprint (See paper)
  • Dual-grained Cache -> Dual-grained Footprint (See paper)

• Evaluation
Partial Footprint

*Time-preserving trace (TP)*

- **Time-preserving trace** is a copy of the original trace with 0 or more accesses replaced with placeholders.

**Trace:**
```
```

**TP:**
```
```
Partial Footprint

Partial Footprint Model

Let $s_i$ be a TP trace, its Partial Footprint is defined as:

$$pfp(x) = \frac{1}{N - x + 1} \sum_{t=x}^{N} \omega(s_i, t, x)$$

TP: $\begin{bmatrix} a[1] & - & - & a[6] & - & - \end{bmatrix}$

$$\text{AVG}(1 \ 1 \ 1 \ 1 \ 1) = 1$$

Let $\Omega = \{d_i\}$ be the set of arrays used by the program. We can define the partial footprint of the TP trace that preserves array $d_i$, denoted as $pfp(d_i, x)$. 

\[ \text{AVG}(1 \ 1 \ 1 \ 1 \ 1) = 1 \]
## Partial Footprint

### Partial Footprint Model

<table>
<thead>
<tr>
<th>Window length</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>≥ 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial Footprint</td>
<td>1/3</td>
<td>3/5</td>
<td>1</td>
<td>4/3</td>
<td>3/2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cache Size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>≥ 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miss Ratio</td>
<td>4/15</td>
<td>2/5</td>
<td>1/3</td>
<td>1/6</td>
<td>1/2</td>
<td>0</td>
</tr>
</tbody>
</table>
Partial Footprint

**All-group Cache Performance**

- The all-group cache miss ratio can be calculated in two ways
  1. Compute the all-group cache miss ratio using the Denning-HOTL conversion.

\[ mr(c) = \text{slope} \]

\[ mr(\Omega, c) = \frac{d}{dx} pfp(\Omega, x) \bigg|_{x=1} \]

\[ m_{fg}(\Omega, x) = c \]
Partial Footprint

All-group Cache Performance

- The all-group cache miss ratio can be calculated in two ways
  1. Compute the all-group cache miss ratio using the Denning-HOTL conversion.
  2. Compute the cache miss ratio for each data item and then sum them up.
Partial Footprint

All-group Cache Performance

• The all-group cache miss ratio can be calculated in two ways
  1. Compute the all-group cache miss ratio using the Denning-HOTL conversion.
     \[
     mr(\Omega, c) = \frac{d}{dx} \left. pfp(\Omega, x) \right|_{pfp(\Omega, x)=c}
     \]
  2. Compute the cache miss ratio for each data item and then sum them up.
     \[
     mr(d_i, c) = \frac{d}{dx} \left. pfp(d_i, x) \right|_{pfp(\Omega, x)=c}
     \]
     \[
     mr(\Omega, c) = \sum_{d_i \in \Omega} mr(d_i, c)
     \]

• The equality of this two solutions shows the Composition Invariance.
Agenda

• Footprint Metric

• Three footprint models

• Evaluation
Evaluation

Setup

• 13 benchmarks from SHOC and CUDA Code Samples

• Traces were collected on Tesla M2075 GPU

• 48KB 6-way L1 Cache and 12 KB, 8-way Texture Cache

• Compared our models with Simulation (Baseline), and two modeling techniques, Set-RD$^1$ and PORPLE$^2$

2. Guoyang Chen, Bo Wu, Dong Li and Xipeng Shen. 2014. PORPLE: An Extensible Optimizer for Portable Data Placement on GPU. *In Proceedings of MICRO.*
## Evaluation

### Experiment Result

<table>
<thead>
<tr>
<th></th>
<th>Simulator</th>
<th>Footprint</th>
<th>PORPLE</th>
<th>Set-RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>100%</td>
<td>99.2%</td>
<td>96.6%</td>
<td>100%</td>
</tr>
<tr>
<td>Cost</td>
<td>14.0</td>
<td>1.6</td>
<td>4.7</td>
<td>524</td>
</tr>
</tbody>
</table>

PORPLE is **111** times faster than Set-RD, and Footprint analysis is **3** times faster than PORPLE.

The arithmetic average across the 13 benchmarks is **0.0%** for Set-RD, **3.4%** error for PORPLE and **0.8%** for Footprint, which is **4.3 times** smaller error than PORPLE.
Summary

• Three models solve Data Non-uniformity, Uneven Address Mapping and Dual Granularity.

![Array Group](image1)
![Associativity](image2)
![Sector Cache](image3)

Predict the per-array cache misses on heterogeneous memory system with all possible cache sizes.

• Experiment results show lower overhead and higher accuracy.
  - 327 times faster than Set-RD, 3 times faster than PORPLE
  - 99.2% accurate, 4.3 times lower error than PORPLE
Thanks for Listening
Any Questions?