Overview

• Traditional cache replacement and insertion policies mainly focus on block reuse.
• Recent literature has proposed cache compression, a promising technique to increase on-chip cache capacity [Pekhimenko et. al., PACT’12].
• In a compressed cache, block size is an additional dimension.
• Observation: The block most likely to be reused soon may no longer be the best block to keep in the cache.
• Key Idea: Use compressed block size in making cache replacement decisions.
• Solution: We propose three mechanisms: Min-LRU, Min-Eviction, and Global Min-Eviction.

Motivation

Problem: How can we maximize cache performance utilizing both block reuse and size?
• No existing policy considers the many varied block sizes and potentials for reuse in making a replacement decision.

We propose compression aware replacement policies.

Example

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Insert</th>
<th>4x</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set 0</td>
<td>4x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>4x</td>
<td>x</td>
</tr>
</tbody>
</table>

Mechanisms

• Policy 1: Min-LRU
  Insight: LRU evicts more blocks than necessary.
  Key Idea: Evict only the minimum number of LRU blocks.

• Policy 2: Min-Eviction
  Insight: Keeping multiple compressible blocks with less reuse may be more valuable than a single uncompressible block of higher reuse.
  Key Idea: Assign a value based on reuse and compressibility to all blocks and on replacement, evict the set of blocks with the least value.

Assigning Values to Block

• Value function: f(block reuse, block size).
• Monotonically increasing with respect to block reuse.
• Monotonically decreasing with respect to block size.
• Plane (see figure) achieves these goals, but is complex to implement in hardware.
• Reuse/Size (see figure) approximates plane and is less complex.
• Probability of reuse predictor: RRIP [Jaleel et. al., ISCA’10] derivative.

Results

• Min-LRU: 1% increase in IPC over LRU.
• Min-Eviction: 3% increase in IPC over LRU.
• IPC increase due to MPKI decrease.

Conclusions

Min-Eviction: a novel replacement policy for the compressed cache.
• Outperforms current state-of-the-art replacement policies.
• First to consider both compressed block size and probability of reuse.
• Simple to implement.

Further Work:

• Global Min-Eviction: a global replacement policy for the compressed decoupled variable way cache that applies similar insight as Min-Eviction.
• Fairness in compressed cache replacement.
• Multi-core evaluation and analysis (see paper): 4% increase in normalized weighted speedup over LRU in heterogeneous workloads.