A Coldness Metric for Cache Optimization

Raj Parihar       Chen Ding†       Michael C. Huang

Dept. of Electrical & Computer Engineering         †Dept. of Computer Science

University of Rochester, Rochester, NY

June 21, 2013
Motivation

- A **hot** concept in program optimization is hotness
  - Program optimization (hot paths), register allocation (hot variables)

- In contrast, cache optimization has to target **cold** data
  - Hot data, due to frequent usage, tend to stay in caches
  - Majority of the misses in last level cache are caused by cold data

- In this paper, we propose
  - A new metric called **coldness** to aid cache optimizations
  - To quantify the coldness variation as the cache size increases
**Definition: Coldness Metric**

For a cache size $c$, the minimal number of “distinct” addresses to target $r\%$ reduction in the miss ratio is defined as **coldness**

$$\text{coldness}(c, r) = (-1) \times \#\text{uniq_addr}$$

![Graph showing coldness metric vs cache size and reuse distance.](image-url)
Coldness Measurement

- Coldness of SPEC 2006 applications as a function of cache size

(a). Distinct addresses accounting for top 10% misses

(b). Distinct addresses accounting for top 50% misses

Classification based on coldness and # of distinct addresses:

- Less cold: h264ref, sphinx2, astar, gobmk, hmmer, dealII, namd
- Highly cold: lbm, bwaves, perlbench, mcf, soplex, sjeng, libq
Coldness Analysis and Insights

- **Cache size** (↑) ⇒ **Coldness** (↓)
  - Coldness for 10% misses: -15 (1KB) ⇒ -4630 (4MB)
  - Coldness for 90% misses: -11K (1KB) ⇒ -50K (4MB)

- **Optimization difficulty** (↑) ⇒ **Coldness** (↓)
  - For a 4MB cache, must optimize accesses to at least 344KB, 2.4MB, and **5.4MB** data to reduce miss ratio by 10%, 50%, and **90%**.

- To achieve 90% reduction in miss ratio for a 4MB cache
  - Must optimize for a data size more than the cache size!
Summary and Future Work

- Majority of the misses are caused by distinct data addresses that originate from a diverse set of distinct instructions.

- Coldness metric successfully quantifies the challenges involved in optimizing (cold) misses for larger caches.

- An effective solution to cold misses is dynamic prefetching. A look-ahead technique, *decoupled look-ahead*, reduces primary misses by 88x for a 4MB cache (with >100K miss addresses).

- Future work: Plan to generalize coldness metric and study software/hardware techniques to leverage it.