An Adaptive, Non-Uniform Cache Structure for Wire-Delay Dominated On-Chip Caches

Summary By: C. Feucht
09/30/2003

1. Notes from Paper

The paper performs an analysis on five different cache architectures: uniform cache (UCA), two static non-uniform caches (S-NUCA-1, S-NUCA-2), multi-level uniform cache (ML-UCA), and dynamic non-uniform cache (D-NUCA). The intent was to show that D-NUCA outperforms all of the other configurations through the use of its various policy decisions.

To determine the best policies to implement, several key areas were identified: mapping, locating blocks, smart search, and replacement. Within these areas several alternatives were examined. A breakdown of these possibilities as well as a few notes about each is listed as follows:

- Mapping – how banks are mapped to sets
  1. Simple – Columns become the ways of the sets (results in speed differences between sets)
  2. Fair – Average access time across the sets is equalized (results in complex routing)
  3. Shared – All back sets share the closest banks
- Locating blocks
  1. Incremental – Step bank by bank in search
  2. Multicast – Send request to some or all of the banks
  3. Hybrid – A mix of incremental and multicast
- Smart Search – Requires some partial tag info kept in array in the cache controller
  1. SS-Performance – Search array and partial tags, and go to memory immediately if the tag isn’t found in the partial tags

2. SS-Energy – Search the closest bank and partial tags in parallel. If closest bank misses, go to the location(s) specified by the partial tag match

- Replacement
  1. Zero-copy – victim line is evicted
  2. One-copy – victim is placed in a lower priority bank , thus displacing less important data

From the analysis done it was found that data promotion had little effect of the IPC and tail eviction was the best overall. Hybrid multicast and SS-energy with shared mapping had the best balance of bank accesses and IPC, so that was characterized as DN-best. This configuration not only outperformed the other schemes, but was found to be within 16% of the theoretical upper-bound in performance.

2. Discussion and Questions

- What is a 2D mesh wormhole? Packet can cut through routing points, so one doesn't have to buffer it. It stays in the link until the end.
- Verification of D-NUCA will be difficult given the policies that were implemented.
- Shared banking introduces associativity on the bank level as well as the bank set level.
- Partial tag update seems to occur by magic. Its mechanism was left out.
- The replace and place strategy wasn't really expanded upon. Is this a pairwise swap? L1s typically use LRU replacement, but this may not be the best for the L2, and effect this is what the promotion frequency based policy for the D-NUCA is.
- It was observed that D-NUCA basically acts like a multi-level cache with exclusion.
- Inclusion was enforced in the ML-UCA configuration and it would be useful to MP data as well.
- In terms of general promotion, does multi-threading get in the way?

- How would invalidated data be handled in D-NUCA? Are they percolated down as well?
- Snooping becomes harder. How does one quickly know if they have the data?
- Overall, coherence wasn’t covered in this paper. TRIPS paper was recommended as a source of possible information.
- How would this be implemented in a CMP? If the processor cores were put around the outside, blocks may not migrate closer to the cores using it if the competing cores are on opposite sides. This could result in the most shared blocks residing in the center. It is also not obvious what is done when a new block is brought in.
- Processor cores most likely wouldn’t be placed along one side of the cache, since there would be additional latency and wiring delay.
- Stacking the cache on top is a novel idea, but the technology is not quite ready. Issues that would need to be addressed are connectivity and heat.
- Data’s migratory behavior needs to be understood ahead of time. This may require the design of a new swap-type policy