TCM Really Lite

Enoch Huang
Sudarshan Seshadri

Design Methodology

This predictor is based on the TCM Memory Scheduler[1]. The general idea presented in this policy is the idea that memory intensive and latency intensive threads should be clustered separately. Latency intensive threads should be prioritized in order to increase overall throughput of the system, as those threads have rare memory accesses but the memory accesses they do have impact the system greatly.

The threads are periodically reclassified as memory intensive or latency intensive based on the count of memory accesses done by each thread. This is done every 1000 cycles and memory access is tracked every thread for classification purposes. A threshold is defined to classify between memory and latency intensiveness; this threshold varied between 2/NUM_THREADS and 6/NUM_THREADS in the paper. To strike a good balance we select 4/NUM_THEADS as the threshold. The number of memory accesses is periodically reset every 1000 cycles to mitigate every thread being classified as bandwidth intensive.

Results

The cache replacement policy achieved a miss rate of 0.382641308 J.s in competition results using config1 and the required workloads.

Tradeoffs

This design does not attempt to implement the more complicated nuances of the TCM scheduler, such as the niceness metric, as it was not obvious how to extract the bank parallelism and row buffer locality figures needed. Furthermore, the MPKI of each thread did not appear to be tracked, which makes it difficult to implement the MPKI-based prioritization of latency intensive threads. However, this results in a fairly simple and small design.

Storage Overhead

This scheduler has minimal data structures involved. The only additional data structures are a 2-bit array keeping track of whether or not each thread is considered latency or bandwidth intensive, and 64*2=128 for the array keeping track of memory access count for each thread. This totals 130 additional bits.

References

[1] Y. Kim, M. Papamichael, O. Mutlu and M. Harchol-Balter, "Thread Cluster Memory Scheduling: Exploiting Differences in Memory Access Behavior," *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, Atlanta, GA, USA, 2010, pp. 65-76, doi: 10.1109/MICRO.2010.51.