Rate-Driven Control of Resizable Caches for Highly Threaded SMT Processors

Sonia Lopez*, Steve Dropsho†, David H. Albonesi‡, Oscar Garnica*, Juan Lanchares*

*Department of Computer Architecture, Universidad Complutense de Madrid, Spain
†School of Electrical and Computer Engineering, Cornell University
‡School of Computer and Communication Sciences, EPFL, Switzerland

Resizable caches can tradeoff capacity for access speed to dynamically match the needs of the workload. In single-threaded cores, resizable caches adapt to the phases of the running application. In Simultaneous Multi-Threaded (SMT) cores the caching needs can vary greatly across the number of threads and their characteristics, thus, offering even more opportunities to dynamically adjust cache resources to the workload.

We demonstrate that the preferred control policy for data cache resizing in a SMT core changes as more threads are run. Prior results on one and two thread workloads showed cache resizing should optimize for cache miss behavior because misses typically form the critical path [1]. In contrast, with many independent threads running, we show optimizing for cache hit behavior has more impact since large SMT workloads have other threads to fill-in during a cache miss. Furthermore, these seemingly diametrically opposed policies are closely related mathematically; the former minimizes the arithmetic mean cache access time, while the latter minimizes its harmonic mean.

Our algorithm, named hybrid algorithm, smoothly and naturally adjusts between the two strategies with the degree of multithreading.

To demonstrate the effectiveness of our proposed cache control algorithm, we implement a quad-threaded core using Simplescalar simulator optimized to run with a small, fast cache that can adjust to greater demands by dynamically upsizing. We adopt the Accounting Cache design of [2] for our resizable caches, but implement a new cache control algorithm that better balances multithreaded needs compared to the original algorithm designed for the single threaded case. The GALS design, in particular, the Multiple Clock Domain (MCD) approach of [3], decouples the adaptive caches from the execution core. Unlike [3], our MCD processor supports SMT and a different domain organization. Our simulation parameters have been chosen to match the characteristics of the Alpha21264, but with additional resources for four threads. Our workload consists of sixteen programs from the spec2000 suite, combined into single, dual and quad-thread workloads.

Our results demonstrate that the new hybrid control algorithm is effective at reacting to single, dual, and quad thread workload phase behavior. Figure 1 summarize the differences between the two algorithms. Whereas the original control algorithm performs well for single and dual thread workloads giving 16.9% and 11.2% improvements over the best synchronous processor baseline, the benefits from optimizing for cache miss behavior, as the original algorithm does, disappear under quad thread loads, averaging -2.5%. In sharp contrast, the proposed hybrid algorithm, which optimizes for cache hit behavior, adjusts well to heavy SMT workloads and generates consistent performance benefits across all workloads tested; specifically, improvements of 16.9%, 16.2%, and 14.2% for 1-, 2- and 4-thread workloads, respectively.

References