An Adaptive Cache Structure for Future High-Performance Systems

Changkyu Kim Doug Burger Stephen W. Keckler

Computer Architecture and Technology Laboratory
Department of Computer Sciences
Tech Report TR-02-10
The University of Texas at Austin
cart@cs.utexas.edu — www.cs.utexas.edu/users/cart

Abstract

On-chip cache sizes are likely to continue to grow over the next decade as working sets, available chip capacity, and memory latencies all increase. Traditional cache architectures, with fixed sizes and discrete latencies, lock one organization down at design time, which will provide inferior performance across a range of workloads. In addition, expected increases in on-chip communication delays will make the time to retrieve data in a cache a function of the data's physical location. Consequently, cache access times will become a continuum of latencies rather than a single one. This non-uniformity will make static organizations particularly limited for single-chip servers, in which multiple processors will be different distances from the cache controller. In this paper, we propose a set of adaptive, high-performance cache design, called Non-Uniform Cache Architectures (NUCAs). We extend these physical designs with logical policies that allow important data to migrate closer to the processor within the same cache. We show that these adaptive level-two NUCA designs provide 1.6 times the performance of a Uniform Cache Architecture of any size, and that the adaptive NUCA scheme outperforms static NUCA schemes by 9% for multi-megabyte, on-chip server caches with large numbers of banks.

1 Introduction

Long memory latencies and limited off-chip bandwidth have driven steady, consistent increases in the sizes of on-chip caches. Processors in late 1980s only included a small level-1 cache (such as the 8KB cache on the first Intel 80486), and these structures grew to 64-128KB in the mid 1990's [19]. Today's high performance processors have continued to increase cache capacities, such as the Alpha 21364 [10] with a 1.5MB L2 cache, and the HP PA-8700 with 2.25MB of combined on-chip cache capacity [12]. The size of cache memories integrated on the processor dies are expected to continue to increase as the bandwidth demands on the package grow [14], as smaller technologies permit more bits per mm^2, and as larger workloads produce correspondingly larger working sets. Demonstrating the likely trend toward even larger on-chip memory systems is the development of large off-chip level-3 caches in today's computers, such as those found in IBM's POWER4 systems [8].

Current multi-level cache hierarchies are organized into a few discrete levels. Typically, each level replicates the contents of the smaller level above it (if inclusion is obeyed), and accesses to the levels in the cache hierarchy are serialized. An access to main memory requires misses in all levels of the hierarchy, assuming that parallel lookups are not used. Cache designers have typically sized the caches so that each successively larger level of cache has an order of magnitude greater capacity and access time. For many applications, the large increase in cache capacity at each level greatly reduces number of misses there and compensates for the added overheads of having the additional level.

In future technologies, large on-chip caches with a single, discrete hit latency will be undesirable, due to increasing global wire delays across the chip [23]. Data residing in the part of a large cache close to the processor could be accessed much faster than data that reside physically farther from the processor. For example, according to our projections the closest bank in a 16-megabyte, on-chip level-two cache built in a 50-nanometer process technology, could be accessed in 5 cycles, while an access to the farthest bank might take 25 cycles. A secondary problem with a monolithic cache architecture is that it can force accesses to be sequentialized, resulting in higher latencies when accesses are pending. While this problem can be mitigated by adding additional ports to the cache, the cost of these ports for a large cache is prohibitive.

Allowing the banks within a cache to be accessed at