Title: Efficient Arbitration Policies For Shared Cache Bandwidth Management

Author: Garima Modi (2018CSZ8010)
Department of Computer Science and Engineering, IIT Delhi

Abstract:

Multiprocessor System-on-Chips (MPSoCs) house multiple processor cores on a single chip, allowing them to execute applications concurrently while sharing memory resources. Their memory hierarchy typically includes private caches (L1, L2), a shared cache (L3), and main memory. Shared caches serve accesses from multiple concurrently executing applications. These shared caches receive *requests* issued by the processor cores. Requests that are subject to cache misses may result in the generation of *responses* received from the lower level of the memory hierarchy. The outstanding requests and responses contend for the shared cache bandwidth, which may degrade the performance of the cores. To manage contention at the shared cache, an arbitration strategy is needed that accounts for the application's bandwidth sensitivity.

Prior research on shared cache management has neglected the additional cache contention caused by responses, which are written to the cache. We propose *CABARRE*, a novel request and response arbitration policy at shared caches, so as to improve the overall system performance. CABARRE ensures run-time management of the shared cache bandwidth by capturing an application's shared cache bandwidth sensitivity and the effect of contention on the memory latency observed by each core. CABARRE shows a performance improvement of 23% on average in a set of SPEC workloads compared to straightforward adaptations of state-of-the-art solutions.

CABARRE, like prior arbitration strategies for shared caches, focuses primarily on improving overall system performance. However, optimizing purely for performance may lead to disproportionate slowdowns of certain applications, resulting in an unfair system. A system is considered fair when all concurrently running applications experience similar levels of slowdown relative to their standalone performance. Therefore, fairness alongside performance must be a key consideration, especially in today's diverse computational landscapes. Arbitration strategies that optimize only performance or only fairness are not sufficient. The former may improve the overall system throughput, but can lead to severe performance degradation of certain applications, resulting in unfair resource allocation. The latter may ensure a balanced resource distribution but might reduce overall system performance. This underscores the need for arbitration policies that strike a balance, maximizing performance while preserving fairness. In our subsequent work, we propose *FARRE*, a novel fairness aware request-response arbitration technique for shared caches. FARRE is designed to optimize performance while attempting to maintain a user-defined fairness threshold. We evaluate its effectiveness through extensive simulations, including comparisons with state-of-the-art arbitration schemes. The results show that FARRE meets or exceeds the input fairness thresholds and improves system performance over standard

fair scheduling policies such as round-robin. Performance improvements reach 14% for lower fairness thresholds, and even at aggressive thresholds, FARRE provides a 5% performance gain. Additionally, compared to the best performance-optimized techniques, FARRE achieves 81% higher fairness.

The aforementioned works focus on demand requests and responses. However, hardware prefetchers are deployed in the system, which speculatively fetch data from memory and place it into caches ahead of demand, so that when a core later requests the data, it avoids the memory latency penalty. Consequently, caches receive two types of requests: *demand requests* from processor cores and *prefetch requests* that proactively bring data closer to the processor. These request types differ in criticality and impact execution differently -- demand requests could be time-critical, as delays can directly stall the processor, whereas timely prefetch responses can reduce the latency of future demand requests and potentially improve performance. Intelligent scheduling of demand and prefetch traffic is therefore essential for obtaining performance gains. Effective scheduling should therefore consider both the type of request and the workload's sensitivity to interference. However, prior work on shared cache bandwidth management is prefetch-oblivious. Treating these request types as equivalent can lead to suboptimal performance. To address this challenge, we introduce *PERCH*, a prefetch-aware request–response arbitration mechanism for shared caches. PERCH differentiates between demand and prefetch requests and distributes cache bandwidth according to each workload's sensitivity to contention, prefetch effectiveness, and the current level of contention in the shared cache. Across a diverse set of 8-core systems running SPEC benchmarks, PERCH achieves average performance gains ranging from 2% to 51% over conventional arbitration policies.

Viva Details:

Date: Friday, 03 July 2026
Time: 11:00 AM – 12:00 PM
Location: ANSK School of IT, SIT 113 (First Floor)