1 Summary

High Performance Cache Replacement Using Re-Reference Interval Prediction (RRIP), 2010: The authors are proposing high-performance (static and dynamic) re-reference interval prediction (RRIP) cache replacement policies. The problem they identified with popular policies at the time of this paper was that none gave enough efficient cache performance across all memory access patterns. Policies that always predict near-immediate RRI (i.e. likely to be referenced again soon) limits cache performance for mixed access patterns (i.e. blocks in the working set are re-referenced sooner (near-immediate RRI) and later (distant RRI have poor temporal locality)); meanwhile, always predicting distant RRI degrades cache performance for access patterns that have mostly near-immediate RRI. In particular, their goal was to design a replacement policy that could preserve the active and frequently referenced working set in the cache, which LRU can't do, while also not contributing significant hardware overhead and could be integrated with the existing LRU cache structures. Their SRRIP uses the concept of re-reference prediction values (RRVP) to introduce long RRI to make better predictions (using two more bits per block) that are statically determined on cache hits and misses (reminiscent of how some OS scheduling policies determine priority between running processes); and their DRRIP uses set dueling in reserved cache sets to estimate between two policies (SRRIP or Bimodal RRIP) is suited for a particular application.

2 Strengths

- In their study, they picked a broad range of memory-intensive and non-memory intensive workloads on single- and multi-cores. Diversity in the workloads strongly supports their findings that their proposed policies are suited across more memory access patterns in different applications than LRU.

3 Weaknesses

- They looked at comparing their proposed policy against a recency-based one (LRU) and compared the SRRIP and DRRIP results against those for NRU, and much of the discussion revolved around comparisons against NRU. If they were available at the time of this paper, I would have liked more of a discussion of SRRIP/DRRIP against more policies, such as other recency-based replacement policies, frequency-based policies like LFU, queue-based policy, Belady-based policies, etc. Section 6.8 does it too briefly, and the paper made it seem like NRU was the only policy comparison they did before section 6.8 mentions a few others.

4 Rating: 4

5 Comments

Overall, they made the approach to their study easy to understand, the analysis and discussions were clear, the experiments replicable, and their introduction and background information for pros and cons found in replacement policies at that time and of what needed to be addressed were detailed and straightforward, setting up a lot of the necessary context for what their SRRIP/DRRIP hoped to tackle, i.e. designing cache replacement policies that required little overhead, could easily be integrated with the existing LRU cache structures in the processors at the time, and more efficiently utilize the cache across or more kinds of memory access patterns across different applications. Much of the analysis and discussion of their results seemed to compare their results and differences in hardware overhead and design changes against typical NRU implementations. For much of the paper, I had thought they only did their study on SRRIP and DRRIP and used NRU as a baseline for comparison, until section 6.8. 6.8 briefly went over a comparison of their SRRIP and DRRIP against a few other policies, using LRU as a baseline. They had some variety of policies in their study (recency-based, frequency-based, queue-based) which strongly supports their findings that SRRIP and DRRIP performed better than those policies available at the time. However, a discussion of the differences in performance was very brief and quickly concluded their favorable findings without delving more into what could have led to the differences in performance and the tradeoffs found within the policies that they compared. For this analysis section, they relied too heavily on the data they presented in just Table 3 and Figure 10, when, even with those two, more of a discussion could have been expanded. I would have liked to have seen more performance analysis of their policies against the other policies with their results in Figures 5-9, as those seemed to focus primarily on just the results of SRRIP. This paper was otherwise interesting to read, especially coming from the perspective of a student where LRU (at a high level) was primarily taught as an example of a good cache replacement policy. I came away from this paper with a better understanding of the kinds of limitations LRU has (ex. unable to distinguish or update re-reference interval predictions between scan and non-scan re-references), what other replacement strategies there are or were being explored, and a solid understanding of how their proposed policies and approach to prediction worked. As an aside, it was interesting to see that SRRIP/DRRIP had an entry on the wiki page for cache replacement policies, and if they are available to include, I think this reading could be accompanied by an example of an architecture that uses SRRIP/DRRIP.