To Cache Or To Bypass? A Fine Balance in The Emerging Memory Technology Era

Kunal Korgaonkar*, Ishwar Bhati, Huichu Liu, Jayesh Gaur, Sasikanth Manipatruni, Sreenivas Subramoney, Tanay Karnik, Steve Swanson*, Ian Young, Hong Wang
UC San Diego*, Intel

1 Abstract

With the availability of new memory technologies like MRAM and ReRAM, the days of SRAM only on-chip caches are likely coming to an end. In our recent work presented at ISCA 2018 [1], we showed the benefits of replacing the SRAMs of Last Level Caches (LLCs) with STT-MRAM in high-performance processors.

Our work unearthed key findings regarding optimal caching/bypassing policies which are unlike those used in current state-of-the-art caching hierarchies. Relative to SRAM, the newer memory technologies can provide 2x to 4x capacity. However, utilizing this capacity to the fullest requires maintaining a fine balance between caching and bypassing. We found that in a hierarchy using emerging technology both caching and bypassing policies become levers to controlling the effective bandwidth (both read and write bandwidth) and the effective latency (affecting the hit rate and hence latency).

As new memory technologies are being introduced, we believe maintaining a balance between caching and bypassing is likely to become even more relevant, not just for on-chip caches, but across the entire memory hierarchy.

2 Proposed Techniques

In the paper, we proposed two main techniques: Write Congestion Aware Bypass (WCAB) and Virtual Hybrid Cache (VHC). These two techniques counter the issues arising when employing STT based caches.

Write Congestion Aware Bypass: One of the ways to reduce LLC congestion, arising due to slow write speeds of STT-RAM, is to bypass some of the writes at the LLC. Many bypassing schemes have been proposed in the context of SRAM LLC. Traditional LLC bypass schemes employ a dead block predictor to classify lines which are less likely to be accessed again and therefore are not filled in the LLC. Bypassing such dead lines retains more live (more likely to be accessed again) cache lines in the LLC and therefore improves hit rate.

In the case of NVM LLC (non-volatile memory technology based LLC), bypassing not only improves hit rate but also reduces write congestion, thereby having a greater impact on performance. Existing schemes adapt SRAM bypass schemes to NVM LLC and demonstrate superior performance. Unfortunately, since these bypass schemes are inherently designed for improving hit rates, the amount of bypass accomplished by them is fairly limited. Moreover, as the LLC capacity grows, the fraction of writes that will not be reused drops as larger capacity, enabled by high NVM density, allows more cache lines with large reuse distances to be retained.

As a result, more aggressive bypass policies are needed to relieve the LLC congestion because of long latency writes. Unfortunately just increasing the aggressiveness of bypass can significantly affect LLC hit rates, thereby negating the capacity benefits offered by the NVM LLC.

We hence need to strike a balance between the conflicting goals of bypassing writes to relieve LLC congestion and the need to minimize the hit rate loss in the LLC because of the bypass.

Given this motivation, our goal for the bypass algorithm is to find an optimal bypass that reduces the LLC queuing latency while minimizing the cost of the bypass.

The optimal bypass depends on request bandwidth demand, the fraction of writes, write latency of STTRAM, main memory latency and the liveness of the application. Of these, the latencies at the LLC and the memory are fixed for a given system design, whereas the liveness, write fraction and request bandwidth need to be learnt dynamically, for a given phase of execution of an application.

In the paper, we describe a write congestion aware bypass (WCAB), that learns these parameters through a simple lightweight learning mechanism, and then uses it to modulate the fraction of bypass. More details about WCAB algorithm [1] can be found in the paper.

Virtual Hybrid Cache: In inclusive LLCs, many cache lines are repeatedly recalled from the LLC, modified in the L2, and written back to the LLC. We call such cache lines that frequently move between the LLC and the L2 as frequent dirty fills.

NVMW'19 Submission, , San Diego, CA, USA 2019.

In exclusive LLC, a read hit deallocates the cache line and moves it to the L2. On an L2 eviction, this line has to be written back to the LLC, irrespective of whether it was clean or dirty. A subsequent hit will move this line back to the L2. Therefore exclusive caches also tend to have a significant amount of frequent clean fills.

To reduce frequent fills at the NVM LLC, we propose a solution that we call as the Virtual Hybrid Cache (VHC).

Unlike true hybrid caches proposed in the literature that use a dedicated SRAM cache for such frequent fills, the VHC simply borrows some capacity from the L2 and the LLC. To reduce writebacks because of dirty L2 evictions, VHC retains cachelines, that create frequent dirty fills, in the L2 so that multiple L1 writebacks may merge in the L2.

For exclusive caches, VHC duplicates some lines in the LLC for reducing frequent clean fills. Recent proposals in the literature also attempt to tackle the clean eviction problem with exclusive caches using a similar approach. However, unlike these proposals, VHC minimizes the LLC capacity loss because of duplication, through smart optimizations in the LLC. More details about VHC [1] are described in the paper.

3 Key results

We next summarize our key results from this work. The experimental setup used to obtain these results is described in the paper.

- Proposed techniques improve the performance (measured as the geometric mean over all 64 traces) of the baseline STTRAM LLC design by 26%. The WCAB component of our policy gains 16% and VHC contributes additionally 10% more performance improvement. Several benchmarks gain significant performance.
- For exclusive LLCs we see that 8MB (2X density) and 16MB (4X density) STTRAM, despite a larger capacity, lose 15% and 13% performance as compared to the 4MB SRAM LLC. However, with our proposals added to the STTRAM LLC, we gain 6% and 12% performance in 8MB and 16MB STTRAM respectively.
- For the inclusive baseline our features gain a significant 11% overall performance.
- On an average, WCAB reduces the writes to the LLC by 11%, while increasing the miss rate by 9%. This trade-off helps it deliver performance. We should note that there are some benchmarks that lose because of WCAB. These benchmarks typically suffer a higher miss rate that is not offset by the improved congestion at the LLC. Overall WCAB performance ranges from -3% to +53%.
- VHC further improves the performance gain by 10%. Overall VHC reduces the number of writes to the LLC and hence helps reduce the aggressiveness of WCAB, thereby improving the overall hit rate.
- We also evaluated our proposal on 15 important industry applications from Enterprise, Database, Big-Data, Desktops, Mobile and HPC domains. Most of these applications run on real world commercial systems and some of them represent emerging usage like machine learning. Overall, STTRAM with our optimization provides 9% performance advantage over the SRAM baseline showing that our proposal is applicable to real-world workloads with a broad range of characteristics.

4 Implications and future work

In our paper, we first showed that LLCs built with emerging NVM memory technologies like STTRAM give sub-optimal performance as compared to SRAM because of long write latency.

We hence proposed a new, low cost, architecture that mitigates this write latency induced performance degradation and improves the performance of a 4 core system with an 8MB of STTRAM based exclusive LLC by an average of 26%. Moreover, we show that the proposed architecture can tolerate high asymmetry in write latency and delivers significant performance improvements over a traditional SRAM LLC.

This can pave the path for the creation of future large LLCs that can effectively utilize the high density offered by these new NVM technologies, while still delivering near SRAM-like performance.

More generally, we believe that the proposed combination of caching and bypass policies may apply to lower levels of the hierarchy as well. Also, while optimality for caching has been a well-studied problem, optimality for a combination of caching and bypassing policies is not well understood. These remain interesting, unexplored problems which we plan to explore as future work.

References

[1] Kunal Korgaonkar, Ishwar Bhati, Huichu Liu, Jayesh Gaur, Sasikanth Manipatruni, Sreenivas Subramoney, Tanay Karnik, Steven Swanson, Ian Young, and Hong Wang. 2018. Density Tradeoffs of Non-volatile Memory As a Replacement for SRAM Based Last Level Cache. In *Proceedings of the 45th Annual International Symposium on Computer Architecture (ISCA '18)*. IEEE Press, Piscataway, NJ, USA, 315–327. https://doi.org/10.1109/ISCA.2018.00035