Persistent Memory: Questions You Should Ask

Tony Mason*
University of British Columbia
Vancouver, BC, Canada
fsgeek@cs.ubc.ca

Abstract

Understanding the performance of byte addressable non-volatile memory (PMEM) is essential to effectively using it across a broad range of applications. Existing work either treats PMEM as a new class of fast storage — due to its persistence — or as a bigger instance of DRAM – due to its increased density. We find the narrowness of these models sacrifice performance. Thus, in our work wee seek to identify and exploring relevant fundamental questions about how PMEM works, and how we adapt to it in order to obtain optimal performance. These questions span both hardware and software architectures and require rethinking how we do things.

Introduction

We seek to identify key questions that impact the effective utilization of byte-addressable persistent memory (PMEM). While PMEM research started decades ago [6] it has accelerated with the April 2019 release of Intel released Optane™ DC Persistent Memory (DCPMEM). Initial work with DCPMEM has focused on raw memory performance and treats PMEM as either high-density DRAM or fast storage [1, 5, 7]. While first generation PMEM is slower than DRAM, future generations will be faster [2]. We find PMEM performance behavior relates to na ive use and prior design decisions that are sub-optimal for PMEM.

We have identified locality as a critical aspect of efficient use of PMEM. Prior locality optimization decisions do not fit with new generation PMEM: processor caches have not kept pace with 10x more dense memories, DAX file systems have not been optimized to preserve PMEM locality, and memory management sizes remain the same as in 1985.

Modern CPUs continue to use 4KB pages that were selected in an era of slow I/O devices — used for demand paging. Linux attempts to address this by defaulting to the largest page sizes possible when applications use persistent memory. Unfortunately, systems tools, such as DAX-aware file systems, unaware that their locality decisions dominate performance in some cases, undo the benefits of those large-page OS enhancements.

Modern CPUs provide robust and efficient support for demand paging. However, when large datasets are stored in persistent memory, there is no benefit from demand paging — the storage and memory are the same. Utilizing huge page tables and translation lookaside buffers (TLBs) is not beneficial for large datasets.

Optimizing data placement for large datasets and computationally intensive workloads, benefits from optimized data placement, as yet another manifestation of locality importance. Existing tools provide some benefits for efficient data placement of standard, well-understood computational kernels on large datasets. We expect to continue our research in this area to enhance existing tools so they better optimize for PMEM, as well as develop new tools for further improving data locality.

Our question of describing how PMEM behave has led us to observing the importance of locality. Locality manifests in a number of key ways, both in the hardware, such as CPU architecture, and software, such as PMEM management policies. Further, we have identified that tools for reorganizing data placement for efficient access during computation can substantially improve the performance of persistent memory. Future work will explore these questions in greater detail by proposing enhanced systems services and tools for optimizing data placement to exploit locality.

References

[1] Joseph Izraelevitz, Jian Yang, Lu Zhang, Juno Kim, Xiao Liu, Amirsaman Memaripour, Yun Joon Soh, Zixuan Wang, Yi Xu, Subramanya R Dulloor, et al. 2019. Basic Performance Measurements of the Intel Optane DC Persistent Memory Module. *arXiv preprint arXiv:1903.05714* (2019).

[2] Dominic Lane and Manus Hayne. 2020. Simulations of Ultralow-Power Nonvolatile Cells for Random-Access Memory. *IEEE Transactions on Electron Devices* (2020).

[3] Vincent Loechner and Doran K Wilde. 1997. Parameterized polyhedra and their vertices. *International Journal of Parallel Programming* 25, 6 (1997), 525–549.

[4] Tony Mason, Thaleia-Dimitra Doudali, Margo Seltzer, and Ada Gavrilovska. 2020. Unexpected Performance of Intel Optane DC Persistent Memory. *IEEE Computer Architecture Letters* (2020).

[5] Ivy B Peng, Maya B Gokhale, and Eric W Green. 2019. System Evaluation of the Intel Optane Byte-addressable NVM. In *Symposium on Memory Systems (MEMSYS19)*.

[6] Michael Wu and Willy Zwaenepoel. 1994. eNVy: a non-volatile, main memory storage system. In *ACM SIGOPS Operating Systems Review*, Vol. 28. ACM, 86–97.

[7] Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steven Swanson. 2019. An Empirical Guide to the Behavior and Use of Scalable Persistent Memory. *arXiv preprint arXiv:1908.03583* (2019).

*Student at UBC, also affiliated with Georgia Institute of Technology, fsgeek@gatech.edu