Understanding and Improving Persistent Transactions on Optane DC Memory

1st Pantea Zardoshti
Lehigh University
USA
zardoshti@lehigh.edu

2nd Michael Spear
Lehigh University
USA
spear@lehigh.edu

3rd Aida Vousoghi
Oracle Corp.
USA
aida.vosoughi@oracle.com

4th Garret Swart
Oracle Corp.
USA
garret.swart@oracle.com

***Abstract*—Storing data structures in high-capacity byte-addressable persistent memory instead of DRAM or a storage device offers the opportunity to (1) reduce cost and power consumption compared with DRAM, (2) decrease the latency and CPU resources needed for an I/O operation compared with storage, and (3) allow for fast recovery as the data structure remains in memory after a machine failure. The first commercial offering in this space is Intel Optane Direct Connect (Optane DC) Persistent Memory. Optane DC promises access time within a constant factor of DRAM, with larger capacity, lower energy consumption, and persistence. We present an experimental evaluation of persistent transactional memory performance, and explore how Optane DC durability domains affect the overall results. Given that neither of the two available durability domains can deliver performance competitive with DRAM, we introduce and emulate a new durability model, called PDRAM, in which the memory controller tracks enough information (and has enough reserve power) to make DRAM behave like a persistent cache of Optane DC memory [1].**

I. Introduction

Intel Optane Direct Connect Persistent Memory (Optane DC) can be thought of as a DRAM alternative that has higher density and lower power consumption, albeit at the cost of higher latency and lower throughput. More exciting is that Optane DC memory can be *persistent*: it can retain its contents for extended periods of time, without requiring any energy to do so. Current Optane DC-based systems can operate in two modes: 1) *Memory Mode* treats DRAM like a cache of the Optane DC memory and disregards persistence. 2) *AppDirect Mode* treats the Optane DC and DRAM as separate memories. To benefit from persistence, it is not enough to simply run an application in AppDirect Mode, because some parts of the system are not persistent. Systems vary in terms of which of their components are persistent, i.e., which are part of the "Durability Domain". There are two persistent domains: in the first domain the Optane memory and the memory controller are persistent: once a store reaches the boundary of the Asynchronous DRAM Refresh (ADR), there is sufficient reserve power to guarantee that the store will pass through the WPQ to the Optane memory and be written, even if the system experiences a power failure. In the second domain, extended ADR ("eADR") provides enough power to allow the operating system to execute instructions that cause all of the data in the caches and WPQ to be flushed to the Optane DIMMs. With eADR, it is not necessary for programs to explicitly execute `clwb` and fence instructions.

In this abstract, we focus on two questions. The first is quite simply "How effectively do measurements on DRAM systems predict performance on Optane DC systems?" The second question is "What is the performance impact of providing enough reserve power to operate in the eADR durability domain?"

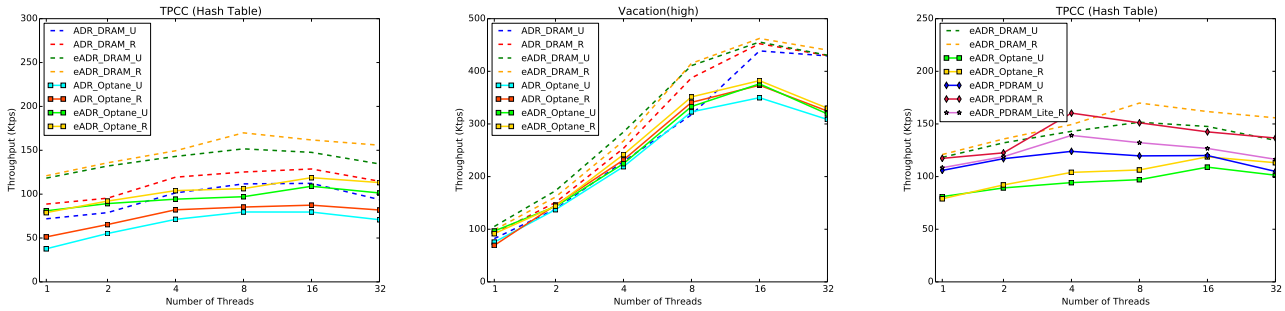
This work has been accepted for publication and presentation at the 34th IEEE International Parallel and Distributed Processing Symposium (IPDPS)

II. PTM Performance on Optane

Experimental Platform: All experiments were conducted on a system containing two 2.30 GHz Intel Xeon Gold 5218 CPUs with 192GB of DRAM and 1.5 TB of Optane DC memory with enabled interleaving. Each CPU has 16 cores / 32 threads, runs Linux kernel version 4.14.35. Experiments are the average of five trials; to avoid NUMA effects, we limited execution to a single CPU socket. Software was compiled using LLVM/Clang 6.0 with O3 optimizations. We used the open-source LLVM PTM plugin [2], which provides a suite of different PTM algorithms [3]. Experiments use the DAX filesystem and Makalu allocator [4] to manage memory from the persistent heap.

Comparing DRAM and Optane Behaviors in ADR: In this subsection, we focus on the four curves marked "ADR" on Figure 1(a)(b). The "U" and "R" suffixes indicate undo logging or redo logging and "Optane" use Optane DC memory in AppDirect mode. Our first finding is that past recommendations regarding the costs of undo logging remain true: in almost every case, redo logging outperforms undo logging. This is despite the higher instruction count for redo logging (due to reads performing lookups in the redo log), and a direct consequence of the cost of fences for undo logging. Moreover, Optane is *worse* than scalability on DRAM. For example, in the Vacation workloads the maximum throughput is reached at a lower thread count, and the gap at peak throughput is substantially larger than the gap at low thread counts. In addition, it is known that Optane DC reads tend to scale with the thread count, whereas writes reach their maximum throughput quickly [5].

Contrasting eADR and ADR Performance: Next, we compare the performance of the system under the ADR and



(a) TPCC (Hash Table) (b) Vacation (high) (c) TPCC (Hash Table)

Fig. 1: Performance comparison between DRAM and Optane (a)(b) and different durability models(c) for workloads.

eADR durability domains. Returning to Figure 1(a)(b), the most significant finding is that eADR provides substantial performance gains for every workload except Vacation. When we focus on the "redo" PTMs, this result speaks to the latency of `clwb` instructions, as they are the only aspect of the algorithm that changes. Clearly, avoiding the need to flush cache lines to the memory controller has a significant impact on performance. Even with these advantages, eADR still does not reach the performance of DRAM. There are two related factors which introduce latency. The first is that the WPQs are bounded, and become saturated. The second is that write latency is higher for Optane than for DRAM. Note that while the eADR PTMs do not explicitly issue `clwb` instructions, data still evicts from the L3 to Optane, through the WPQs. The known problem of WPQ saturation [5] explains the decrease in scalability.

III. New Models for Persistence

In Section II, we observed that eADR can substantially improve performance versus ADR, primarily because eADR does not require explicit fences and flushes. In this section, we introduce two new durability models, which are able to deliver better performance than eADR. While neither is available in hardware today, nor does either require substantially different support than is available in Optane DC systems today. The fundamental enabling mechanism for our new durability models is the directory used by the memory controller when the system runs in Memory Mode.

The Persistent DRAM Durability Domain: Our first new durability domain, PDRAM, gives the illusion that all of DRAM is persistent. It combines the persistence of AppDirect Mode with the caching behavior of Memory Mode. Like eADR, PDRAM treats the caches as persistent. However, it requires a directory in DRAM, so that it can potentially flush all of DRAM to Optane on a power signal.

The PDRAM-Lite Durability Domain: We note that making *all* of DRAM into a cache of Optane memory may not be advantageous. On the one hand, certain memory regions (such as the stack, or the lookup tables of a redo log) typically do not require persistence. Additionally, the specific case of redo-based PTM has simpler persistence requirements than undo-based PTM.

Notice that in redo-based PTM, a transaction only performs stores to the Optane memory *at commit time*. Until the commit point, a redo-based transaction keeps its entire write working set in the (highly compact) redo log. For transactions with modest write set sizes, it would be possible to use a small amount of PDRAM for the transactions' redo logs, without caching any other Optane pages in DRAM. We refer to this approach as PDRAM-Lite.

Evaluation of PDRAM and PDRAM-Lite: Figure 1(c) repeats the experiments of Section II. The first goal of these experiments is to determine whether PDRAM can bridge the gap between the DRAM and eADR configurations. The result is largely affirmative. PDRAM matches DRAM performance up until Optane scalability bottlenecks (e.g., WPQ saturation) occur.

The second goal of these experiments is to determine whether PDRAM-Lite offers sufficient value. The result here is less clear: on the one hand, PDRAM-Lite outperforms eADR in every case. In fact, this result confirms a finding from [5]: Optane DC© throughputs are much closer to DRAM throughput for regular access patterns than for irregular patterns. Moving the redo log into DRAM does not have a significant impact on latency, since the compact log, with its regular access pattern, did not have much worse latency than DRAM to begin with.

References

[1] P. Zardoshti, M. Spear, A. Vousoghi, and G. Swart, "Understanding and Improving Persistent Transactions on Optane DC Memory," in *Proceedings of the 34th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, New Orleans, LA, May 2020.

[2] P. Zardoshti, T. Zhou, P. Balaji, M. L. Scott, and M. Spear, "Simplifying Transactional Memory Support in C++," p. 25, 2019.

[3] P. Zardoshti, T. Zhou, Y. Liu, and M. Spear, "Optimizing Persistent Memory Transactions," in *Proceedings of the 28th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, Seattle, WA, Sep. 2019.

[4] K. Bhandari, D. R. Chakrabarti, and H.-J. Boehm, "Makalu: Fast recoverable allocation of non-volatile memory," in *ACM SIGPLAN Notices*, vol. 51, no. 10. ACM, 2016, pp. 677–694.

[5] J. Izraelevitz, J. Yang, L. Zhang, J. Kim, X. Liu, A. Memaripour, Y. J. Soh, Z. Wang, Y. Xu, S. R. Dulloor *et al.*, "Basic Performance Measurements of the Intel Optane DC Persistent Memory Module," 2019.