An Empirical Guide to the Behavior and Use of Scalable Persistent Memory

Jian Yang Juno Kim Morteza Hoseinzadeh Joseph Izraelevitz[†] Steven Swanson
University of California, San Diego [†]University of Colorado, Boulder

1 INTRODUCTION

We have characterized the performance and behavior of Optane DIMMs using a wide range of micro-benchmarks, benchmarks, and applications. The data we have collected demonstrate that many of the assumptions that researchers have made about how NVDIMMs would behave and perform are incorrect. We have found the actual behavior of Optane DIMMs to be more complicated and nuanced than the "slower, persistent DRAM" label would suggest. This paper presents a detailed evaluation of the behavior and performance of Optane DIMMs on microbenchmarks and applications and provides concrete, actionable guidelines for how programmers should tune their programs to make the best use of these new memories.

2 OPTANE MEMORY

The Optane Memory (or Optane DIMM) is the first scalable, commercially available NVDIMM. It has lower latency, higher read bandwidth than existing storage devices, and presents a memory address-based interface. Compared to DRAM, it has higher density and persistence.

Optane DIMM sits on the memory bus, and connects to the processor's integrated memory controller (iMC). On the platform that supports Optane DIMMs, each processor contains one or two processor dies which comprise separate NUMA nodes. Each processor die has two iMCs, and each iMC supports three channels. Therefore, in total, a processor die can support a total of six Optane DIMMs across its two iMCs.

Stores are considered persistent when they reach the *asynchronous DRAM refresh (ADR)* domain inside the iMC. The iMC communicates with the Optane DIMM using the DDR-T interface in cache-line (64-byte) granularity. Memory accesses to the NVDIMM arrive first at the on-DIMM controller (*XPController*), which coordinates access to the Optane media. Once the address translation is performed, the actual access to storage media occurs. As the Optane physical media access granularity is 256 bytes (*XPLine*), the XPController will translate smaller requests into larger 256-byte accesses, causing write amplification as small stores become read-modify-write operations. The XPController has a small write-combining buffer (*XPBuffer*) to merge adjacent writes.

3 PERFORMANCE CHARACTERIZATION

We measure Optane's performance along multiple axes and find that its performance characteristics are complex and surprising in many ways, especially relative to the notion that Optane behaves like slightly-slower DRAM.

Latency Read and write latencies are key memory technology parameters. We measure read latency by timing the average latency for 8-byte load instructions to sequential and random memory addresses. To eliminate caching and queuing effects, we empty the CPU pipeline and issue a memory fence (`mfence`) between measurements. For writes, we load the cache line into the cache and then measure the latency of one of two instruction sequences: a 64-bit store, a `clwb`, and an `mfence`; or a non-temporal store followed by an `mfence`.

Our results (Figure 1 (c)) show the read latency for Optane is 2×–3× higher than DRAM. We believe most of this difference is due to Optane's longer media latency. Optane memory is also more pattern-dependent than DRAM. The random-vs-sequential gap is 20% for DRAM but 80% for Optane memory, and this gap is a consequence of the XPBuffer. For stores, the memory store and fence instructions commit once the data reaches the ADR at the iMC, so both DRAM and Optane show a similar latency. Non-temporal stores are more expensive than writes with cache flushes (`clwb`).

Bandwidth Detailed bandwidth measurements are useful to application designers as they provide insight into how a memory technology will impact overall system throughput. Figure 1 (a) shows the bandwidth achieved at different thread counts for sequential accesses with 256 B access granularity while Figure 1 (b) shows the bandwidth achieved with varying access size. For the latter, we use the best-performing thread count each curve given as "<load threads>/ <ntstore threads>/<store+clwb threads>". We show loads, non-temporal stores, and cached writes with flushes. In both graphs, the left-most graph plots performance for interleaved DRAM, while the center and right-most graphs plot performance for non-interleaved and interleaved Optane. In the non-interleaved measurements all accesses hit a single DIMM.

The data shows that DRAM bandwidth is both higher than Optane and scales predictably (and monotonically) with thread count until it saturates the DRAM's bandwidth and that bandwidth is mostly independent of access size.

The results for Optane are wildly different. First, for a single DIMM, the maximal read bandwidth is 2.9× of the maximal write bandwidth (6.6 GB/s and 2.3 GB/s, respectively), where DRAM has a smaller gap (1.3×). Second, with the exception of interleaved reads, Optane performance is non-monotonic with increasing thread count. For the non-interleaved (i.e., single-DIMM) cases, performance peaks at between one and four threads and then tails off. Interleaving pushes the peak to twelve threads for `store+clwb`. Third, Optane bandwidth for random accesses under 256 B is poor. This "knee" corresponds to XPLine size.

4 BEST PRACTICES FOR OPTANE MEMORY

The basic differences between Optane and conventional memory technologies mean that existing intuitions about how to optimize software for disks and memory do not apply directly to Optane. We

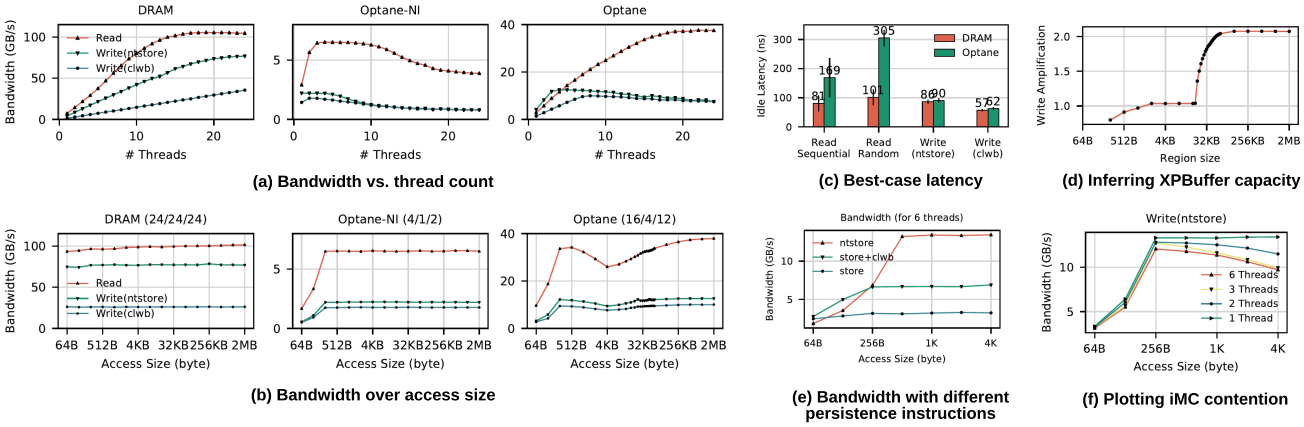

Figure 1: Overview of the Optane DIMM Performance

distill the results of our characterization experiments into a set of four principles for how to build and tune Optane-based systems.

Avoid small random accesses. Internally, Optane DIMMs update Optane contents at a 256 B XPLine granularity, causing write amplifications for smaller updates due to an internal read-modify-write operation. To compensate this, Optane DIMM has XPBuffer, a small write-combining buffer that buffers and combines 64 B accesses into 256 B internal writes. As a consequence, Optane DIMMs can efficiently handle small stores, if they exhibit sufficient locality.

To understand how much locality is sufficient, we crafted an experiment to measure the size of the XPBuffer: First, we allocate a contiguous region of N XPLines. During each "round" of the experiment, we first update the first half (i.e., 128 B) of each XPLine in turn, and then update the second half of each XPLine. Figure 1 (d) shows that the amplification ratio significantly jumps at $N = 64$ (a region size of 16 kB), indicating a sharp rise in the miss rate for the second half accesses. This result implies the XPBuffer is approximately 16 kB in size.

These results provide specific guidance for maximizing Optane store efficiency: Avoid small stores, but if that is not possible, limit the working set to 16 kB per Optane DIMM.

Use non-temporal stores for large writes. The choice of how programs perform and order updates to Optane has a large impact on performance. Programmers have several options: After a regular `store`, they can either evict (`clflush`, `clflushopt`) or write back (`clwb`) the cache line to move the data into the ADR and eventually the Optane DIMM. Alternatively, the `ntstore` writes directly to persistent memory, bypassing the cache hierarchy. For all these instructions, a subsequent `sfence` ensures that the effects of prior evictions, write backs, and non-temporal stores are persistent.

In Figure 1 (e), we compare achieved bandwidth for sequential accesses with three different instruction sequences: `ntstore`, `store + clwb`, and `store`, followed by a `sfence`. The data show that flushing after each 64 B store improves the bandwidth for accesses larger than 64 B. We believe this occurs because letting the cache naturally evict cache lines adds nondeterminism to the access stream that reaches the Optane DIMM. Proactively cleaning the cache ensures that accesses remain sequential. The data also show that non-temporal stores have highest bandwidth for accesses over 256 B. Here, the performance boost is due to the fact that a `store + clwb` must load the cache line into the CPU's local cache before executing store, thereby using up some of the Optane DIMMs bandwidth. As `ntstores` bypass the cache, they will avoid this extraneous read and can achieve higher bandwidth.

Limit the number of concurrent threads accessing an Optane DIMM. Systems should minimize the number of concurrent threads targeting a single DIMM simultaneously. An Optane DIMM's limited store performance and limited buffering at the iMC and on the DIMMs combine to limit its ability to handle accesses from multiple threads simultaneously. We have identified two distinct mechanisms that contribute to this effect.

First, contention for space in the XPBuffer leads to increased evictions and write backs to Optane media, driving up the write amplification. The center figure in Figure 1 (a) shows this effect.

Second, limited queue capacity in the iMC also hurts performance when multiple cores target a single DIMM. Figure 1 (f) shows an experiment that uses a fixed number of threads to write data to 6 interleaved Optane DIMMs. We let each thread access N DIMM (with even distribution across threads) randomly. As N rises, the number of writers targeting each DIMM grows, but the per-DIMM bandwidth drops. A possible culprit is the limited capacity of the XPBuffer, but the write amplification remains very close to 1, so the performance problem must be in the iMC. Our hypothesis is that, since Optane DIMMs are slow, they drain the write-pending queue in the iMC slowly, leading to head-of-line blocking effects.

Avoid NUMA accesses. NUMA effects for Optane are much larger than they are for DRAM, so designers should strive to avoid cross-socket memory traffic. Our experiments reveal that remote Optane accesses show 1.2×–2.5× slowdown (depending on the access pattern) when workloads are either read- or write-only. However, the cost is drastically degraded when either the thread count increases or the workload is read/write mixed. Based on the results from our systematic sweep, the bandwidth gap between local and remote Optane memory for the same workload can be over 30×, while the gap between local and remote DRAM is, at max, only 3.3×.