Exploiting Intel Optane SSD for Microsoft SQL Server

Kan Wu, Andrea Arpaci-Dusseau
Remzi Arpaci-Dusseau
University of Wisconsin-Madison

Rathijit Sen
Kwanghyun Park
Gray Systems Lab
Microsoft Corporation

ABSTRACT

New NVM-based devices provide unparalleled performance (i.e., significantly reduced latency) than Flash-based SSDs. In this paper, we look into exploiting an NVM-based block device – the Intel Optane SSD – as a caching layer for Microsoft SQL Server. We reveal that naive usage of Optane SSD can result in up to 23% higher query response time than Flash SSD. We explain the issues of simple caching by analyzing the I/O characteristics of Intel Optane SSD. To exploit Optane SSD as a caching layer, we propose an Optane SSD-aware caching strategy including an optimized cache replacement policy and a cache access filter.

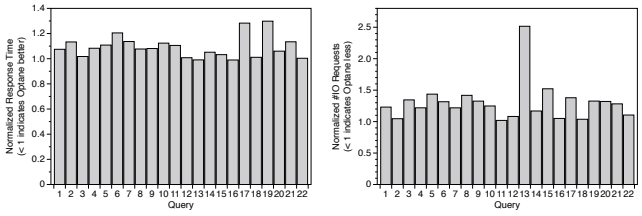
ACM Reference Format:
Kan Wu, Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, Rathijit Sen, and Kwanghyun Park. 2019. Exploiting Intel Optane SSD for Microsoft SQL Server. In *International Workshop on Data Management on New Hardware (DaMoN'19), July 1, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3329785.3329916

1 INTRODUCTION

The recent advance of NVM has yielded storage devices with unparalleled performance levels. For example, Intel Optane SSD [11] provides up to 10x lower latency than classic Flash-based SSD. Intel Optane SSD is constructed from 3D XPoint memory [3, 4] from Intel and Micron. There are various form factors of devices on the market based on 3D XPoint memory, including Optane memory [10] and Optane DC Persistent Memory [9]. Among these devices, the Optane SSD is the most cost-effective option.

Intel Optane SSD can perform a potential role as a caching layer between DRAM and Flash SSD for applications. Intel provides Memory Direct Technology (IMDT) [8] for caching on Optane SSD. According to Intel, there are dozens of use cases of IMDT and Optane SSD. Important examples include Memcached [5], Redis [6], and Spark [7]. While Optane SSD is around 4x [11] more expensive than Flash SSD, it is more cost-effective to keep Flash-based SSD as the primary storage and Optane SSD as a caching layer. For instance, Optane SSDs are deployed to support key workloads in Facebook [1, 2], both as a caching layer between DRAM and Flash SSD for RocksDB and for machine learning.

We look into exploiting Optane SSD as a caching layer for Relational Database Management Systems (RDBMS). RDBMS sizes, especially for analytic workloads, tend to be large [12]. Hence, cost-effective Flash SSDs are favored to be the primary storage. Meanwhile, RDBMS users usually demand fast query response times and high system throughput. Thus, we believe that Optane SSD as a caching layer is a proper configuration for RDBMS. Specifically, we evaluate this configuration in the context of Microsoft SQL Server [12]. Unlike IMDT, which requires high-end CPU and datacenter-level Optane SSD, we propose a flexible solution. The caching logic works at the block layer; accesses to Flash SSD are intercepted and directed to Optane SSD on a cache hit. The caching requires no modification to SQL Server.



(a) Response Time **(b) Normalized Number of I/O Requests**

Figure 1: Normalized TPC-H Response Time and Number of I/O Requests, Optane SSD vs. Flash SSD

Through our initial experience, we observe the inefficiency of naive caching on Optane SSD for SQL Server. For the TPC-H benchmark, Optane SSD results in worse query response time as compared to Flash SSD. After an investigation, we found that the problem is due to I/O characteristics of Optane SSD. Motivated by what is observed, we propose an Optane-aware caching strategy. Especially, we introduce a cache replacement policy that accounts for the specific performance characteristics of the Optane and underlying Flash SSD; a naive approach, which assumes that the newer device is simply faster, no longer will suffice. We also introduce a cache access filter that selectively reduces the load on the Optane, thus ensuring it operates with high performance.

We now present the problems when naively using Optane SSD as a caching layer for Microsoft SQL Server, provide insights to the problems through a discussion, and propose an Optane SSD-aware caching strategy.

2 OPTANE PERFORMANCE

We show the problems of naively caching on Optane SSD through examining the TPC-H benchmark on Microsoft SQL Server 2019. We compare SQL Server's performance in case of 1) all dataset on Optane 905P SSD (960GB) and 2) direct use of a Flash SSD (Samsung 970 Pro 1TB [13]). We choose the TPC-H benchmark with 300SF. Experiments were run on a workstation with Intel Xeon E5 CPU at 2.1GHz (16 cores) using 64GB of DRAM, running Ubuntu 16.04. We report the query response time in a cold cache setting.

DaMoN'19, July 1, 2019, Amsterdam, Netherlands

ACM ISBN 978-1-4503-6801-8/19/07…$15.00
https://doi.org/10.1145/3329785.3329916

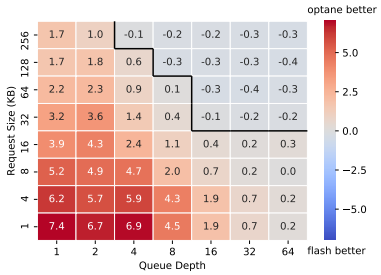


Figure 2: Normalized Difference between Avg Latency of Random Reads, Optane SSD vs. Flash SSD

Separate from what we expect of 3D XPoint memory's low access latency, SQL Server on Flash SSD achieves lower response time for 20 out of 22 TPC-H queries than when it's running on Optane SSD. As shown in Figure 1, the response time on Flash can be up to 23% lower than that on Optane. Optane only outperforms Flash for query 13 and 16, and the improvement is within 3%. The relatively worse behavior of Optane SSD is due to its I/O characteristics.

First, Optane SSD serves large I/O requests inefficiently. As in Figure 1, SQL Server running on Optane issues more block requests compared to its performance on Flash. This I/O amplification on Optane can be up to 2.5x and arises due to Optane SSD's limit on the size of individual block I/O (128KB). The Flash SSD we tested has a much larger maximum block request size (2048KB). The hardware limit of Optane SSD means that it is not suitable for large accesses (>128KB); each large access will be split, resulting in higher latency.

Second, Optane SSD does not improve I/O performance when there is high I/O concurrency. We see constantly lower I/O throughput and higher average access latency from Optane compared to Flash when serving TPC-H queries. The reason for Optane SSD's low I/O throughput is two-fold. First, TPC-H workloads generate a high volume of in-flight I/Os. Most queries are enough to exploit the bandwidth of both Optane and Flash SSD. Furthermore, Optane SSD has limited internal parallelism [14] compared to classic Flash SSD; it provides lower maximum throughput (2500MB/s) than the Flash SSD we tested (3500MB/s). Hence, Optane SSD presents relatively poor performance when serving TPC-H queries.

As for latency, we investigate the reason for the worse behavior of Optane SSD through microbenchmarks, comparing the random read performance of the two devices. For each workload, we vary the size of requests and the parallelism (queue depth) of accesses. From Figure 2, Optane SSD outperforms Flash up to 7.4x for the workload with low request scale (left-down regime), while it does not show improvement for high request scale workloads. This observation suggests that to exploit the latency benefit from Optane SSD, we need to control the access load (request sizes and queue depth) to Optane SSD.

Discussion: The problems we observed suggest a different storage hierarchy than the conventional one. Previously, in a hierarchy including DRAM, SSD, and HDD, there is a total order of storage layers in terms of performance (i.e., **DRAM > SSD > HDD**). Hence, the accesses have a total ordered preference for devices. Presently, there is no longer a total order among layers, especially between Optane SSD and Flash SSD; Optane has its advantages, as well as disadvantages compared to Flash (i.e. **DRAM > Optane** >/=/< **Flash**). This lack of total order between Optane and Flash requires the system to split accesses to the device that best fits them.

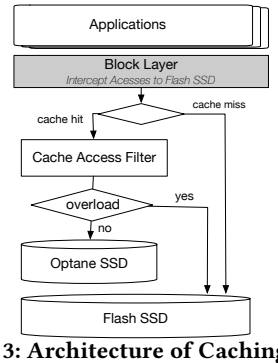


Figure 3: Architecture of Caching Layer

To divide accesses, partitioning the external data structure to devices can be helpful. Using Optane as a caching layer is actually an automatic partitioning method. Over time, the frequently-accessed and Optane SSD-favorable accesses should be cached. To realize this, an Optane SSD-aware caching strategy is needed.

To note, according to [14], the limitations of current Optane SSD are mostly related to its controller/interconnect design. The limitations need to be verified once new NVM-based devices available.

3 OPTANE SSD-AWARE CACHING STRATEGY

We propose caching on Optane SSD for SQL Server with the following caching strategy (Figure 3). Optane SSD behaves as an inclusive, write-through cache. The caching strategy is device-aware in two aspects: the cache-replacement policy and the cache-access filter.

Cache Replacement Policy: An item will be evicted from the Optane SSD according to both locality and the I/O characteristics of the accesses. As in Figure 2, caching a frequently accessed block in Optane may not benefit the application if it is large or if many parallel requests are on-going. Hence, we propose to combine Optane-aware features of each request into the base replacement algorithm (e.g., LRU). These features include request size and historical I/O parallelism (e.g., queue depth) when issuing the request. When we need to free space from Optane SSD, the cached items that are worst along these axes are evicted.

Cache Access Filter: We also need a cache access filter to reduce burst loads to the Optane. The benefit a cache hit can produce depends on the concurrent I/Os to Optane SSD. When there is a burst load, the latency of Optane SSD can be worse than directly accessing Flash. Therefore, we introduce a cache-access filter atop Optane. The filter monitors the number of in-flight I/Os; when this number exceeds a threshold, further cache requests will be directed to Flash instead. The threshold is related to various parameters, including the number and size of in-flight I/Os and the performance difference between Optane and underlying Flash SSD.

4 CONCLUSION & OUTLOOK

We present the problems when naively using Optane SSD as a caching layer for Microsoft SQL Server. We provide insights into the problems and propose an Optane SSD-aware caching strategy. We are working on the implementation of this strategy. We believe that our initial results and analysis are relevant for users of Intel Optane SSD in different application scenarios.

REFERENCES

[1] Assaf Eisenman, Darryl Gardner, Islam AbdelRahman, Jens Axboe, Siying Dong, Kim Hazelwood, Chris Petersen, Asaf Cidon, and Sachin Katti. 2018. Reducing DRAM footprint with NVM in facebook. In *Proceedings of the Thirteenth EuroSys Conference*. ACM, 42.

[2] Assaf Eisenman, Maxim Naumov, Darryl Gardner, Misha Smelyanskiy, Sergey Pupyrev, Kim Hazelwood, Asaf Cidon, and Sachin Katti. 2018. Bandana: Using non-volatile memory for storing deep learning models. *arXiv preprint arXiv:1811.05922* (2018).

[3] Frank T Hady, Annie Foong, Bryan Veal, and Dan Williams. 2017. Platform storage performance with 3D XPoint technology. *Proc. IEEE* 105, 9 (2017), 1822–1833.

[4] Intel. 2019. 3D XPoint. https://en.wikipedia.org/wiki/3D_XPoint.

[5] Intel. 2019. IMDT use case, memcached. https://www.intel.com/content/www/us/en/support/articles/000026359/memory-and-storage/data-center-ssds.html.

[6] Intel. 2019. IMDT use case, redis. https://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/imdt-solution-brief-in-memory-data-store.pdf.

[7] Intel. 2019. IMDT use case, Spark. https://www.intel.com/content/www/us/en/software/apache-spark-optimization-technology-brief.html.

[8] Intel. 2019. Intel Memory Drive Technology. https://www.intel.com/content/www/us/en/software/intel-memory-drive-technology.html.

[9] Intel. 2019. Intel Optane DC Persistent Memory. https://www.intel.com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory.html.

[10] Intel. 2019. Intel Optane Memory. https://www.intel.com/content/www/us/en/architecture-and-technology/optane-memory.html.

[11] Intel. 2019. Intel Optane SSD. https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds/optane-dc-p4800x-series.html.

[12] Microsoft. 2019. Microsoft SQL Server. https://www.microsoft.com/en-us/sql-server/default.aspx.

[13] Samsung. 2019. Samsung 970 Pro. https://www.samsung.com/semiconductor/minisite/ssd/product/consumer/970pro/.

[14] Kan Wu, Andrea Arpaci-Dusseau, and Remzi Arpaci-Dusseau. 2019. Towards an Unwritten Contract of Intel Optane SSD. In *11th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 19)*. USENIX Association, Renton, WA.