A Framework for Large Dynamic Graph Analysis on Persistent Memory

Abdullah Al Raqibul Islam Dong Dai
Computer Science Department, University of North Carolina at Charlotte

Graph-structured data analysis has been extensively used in many real-world applications. As real-time data is fast becoming the normal [1], many of these graphs become dynamic and evolve over time. It is then critical to be able to store the dynamic updates continuously and persistently while, at the same time, executing iterative graph algorithms in real time.

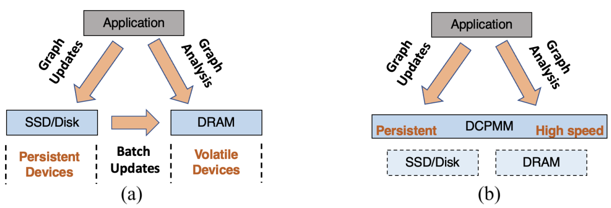

Figure 1: Workflow of dynamic graph frameworks.

Currently, supporting both persistent graph updates and real-time graph analytics needs to properly manage two different types of (persistent and volatile) storage devices. As Fig. 1(a) shows, the *persistent* devices serve *graph updates* for data safety; the *volatile* devices serve *graph analytic* for maximal performance. Since data is written and read in different devices, *batch updates* are needed. Such a divided framework leads to multiple design issues. First, the *batch updates* are expensive and hence should be executed less. However, its execution frequency also determines how well graph analysis can catch up with the latest graph changes, creating a design dilemma for the developers. Second, maintaining data at both locations wastes the storage and makes smaller ones often the throttling factor for large graphs.

Recently, a new set of non-volatile or persistent memory devices, such as Intel Optane DC Persistent Memory (PMEM), emerged [2]. Compared with DRAM, these devices provide data persistence and higher density. Compared with block-based persistent devices, they can be directly accessed in bytes with lower latency and higher IOPS [3–6]. These advanced features open new design spaces for addressing dynamic graph analysis problems. As shown in Fig. 1(b), PMEM can replace both persistent and volatile storage devices to avoid data synchronization and movement issues.

Our Approach

To design an efficient dynamic graph framework on Persistent Memory, we propose to leverage recent progress in Packed Memory Array-based mutable CSR (compressed sparse row) graph structure [7, 8]. PMA-based CSR essentially replaces its edge array using a gaped array to efficiently support both graph updates and analytic.

Due to the unique features of PMEM, naive porting PMA-based CSR to PMEM leads to problematic performance. First, PMA-based CSR introduces frequent data shifts in a small range, which could be extremely inefficient on PMEM as its performance relies on efficiently using the internal 256-byte write buffers [9]. Second, in-place updates on PMEM are known to be extremely slow [10]. But most of the metadata updates to PMA-based CSR are in place. Third, the crash consistency guarantee could be very expensive [11–15] and tricky to implement for many core PMA-based CSR operations.

Graph Operations	Our Solution	GraphOne	LLAMA
Dynamic Insertion	1167.64	2985.07	1348.51
PageRank	545.92	775.83	712.73

Table 1: Performance of different frameworks on PMEM (seconds).

We propose three key approaches to address these issues. First, we introduce *per-segment persistent logs* to reduce unnecessary data shifts. Second, we introduce *per-thread undo logs* to guarantee crash consistency efficiently. Third, we design new *data placement schema* to maximally avoid the in-place data updates on PMEM. Our initial results are promising compared to existing state-of-the-art persistent graph processing systems running on PMEM, such as GraphOne [16] and LLAMA [17]. Table 1 lists the dynamic graph insert time and PageRank runtime on Twitter [18] graph[1]. Our proposed solution can achieve up to 2.56× better performance in dynamic graph insertion and reduce $\sim$ 30% of PageRank time.

[1]Experiment setup: 2nd Gen. Intel Xeon Scalable (Gold 6254 @ 3.10G), 6 DRAM/PMEM DIMMS, 192GB DRAM, 768GB PMEM, Ubuntu 20.04.

References

[1] Tyler Akidau, Robert Bradshaw, Craig Chambers, Slava Chernyak, Rafael J Fernández-Moctezuma, Reuven Lax, Sam McVeety, Daniel Mills, Frances Perry, Eric Schmidt, et al. The dataflow model: a practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. 2015.

[2] Optane. Intel Optane Persistent Memory. `https://www.intel.com/content/www/us/en/products/docs/memory-storage/optane-persistent-memory/optane-dc-persistent-memory-brief.html`, 2019. Accessed: 2023-01.

[3] Kosuke Suzuki and Steven Swanson. A survey of trends in non-volatile memory technologies: 2000-2014. In *2015 IEEE International Memory Workshop (IMW)*, pages 1–4. IEEE, 2015.

[4] Moinuddin K Qureshi, John Karidis, Michele Franceschini, Vijayalakshmi Srinivasan, Luis Lastras, and Bulent Abali. Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling. In *2009 42nd Annual IEEE/ACM international symposium on microarchitecture (MICRO)*, pages 14–23. IEEE, 2009.

[5] J Joshua Yang, Dmitri B Strukov, and Duncan R Stewart. Memristive devices for computing. *Nature nanotechnology*, 8(1):13, 2013.

[6] Moinuddin K Qureshi, Vijayalakshmi Srinivasan, and Jude A Rivers. Scalable high performance main memory system using phase-change memory technology. In *Proceedings of the 36th annual international symposium on Computer architecture*, pages 24–33, 2009.

[7] Brian Wheatman and Helen Xu. Packed compressed sparse row: A dynamic graph representation. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2018.

[8] Abdullah Al Raqibul Islam, Dong Dai, and Dazhao Cheng. Vcsr: Mutable csr graph format using vertex-centric packed memory array. In *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 71–80, 2022.

[9] Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steven Swanson. An empirical guide to the behavior and use of scalable persistent memory. In *Proceedings of the 18th USENIX Conference on File and Storage Technologies*, FAST'20, page 169–182, USA, 2020. USENIX Association.

[10] Youmin Chen, Youyou Lu, Fan Yang, Qing Wang, Yang Wang, and Jiwu Shu. Flatstore: An efficient log-structured key-value storage engine for persistent memory. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20, page 1077–1091, New York, NY, USA, 2020. Association for Computing Machinery.

[11] Swapnil Haria, Mark D Hill, and Michael M Swift. Mod: Minimally ordered durable datastructures for persistent memory. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 775–788, 2020.

[12] Abdullah Al Raqibul Islam, Dong Dai, Anirudh Narayanan, and Christopher York. A performance study of optane persistent memory: From indexing data structures' perspective. In *MSST '20: 36th International Conference on Massive Storage Systems and Technology*, 2020.

[13] Abdullah Al Raqibul Islam and Dong Dai. Understand the overheads of storage data structures on persistent memory. In *Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '20, page 435–436, New York, NY, USA, 2020. Association for Computing Machinery.

[14] Kai Wu, Jie Ren, Ivy Peng, and Dong Li. {ArchTM}:{Architecture-Aware}, high performance transaction for persistent memory. In *19th USENIX Conference on File and Storage Technologies (FAST 21)*, pages 141–153, 2021.

[15] Abdullah Al Raqibul Islam, Christopher York, and Dong Dai. A performance study of optane persistent memory: from storage data structures' perspective. *CCF Transactions on High Performance Computing*, Sep 2022.

[16] Pradeep Kumar and H Howie Huang. Graphone: A data store for real-time analytics on evolving graphs. In *17th {USENIX} Conference on File and Storage Technologies ({FAST} 19)*, pages 249–263, 2019.

[17] Peter Macko, Virendra J Marathe, Daniel W Margo, and Margo I Seltzer. Llama: Efficient graph analytics using large multiversioned arrays. In *2015 IEEE 31st International Conference on Data Engineering*, pages 363–374. IEEE, 2015.

[18] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, 2010.