Emerging Storage Technologies: A Brief Overview

Abir Mahmud
Department of Electrical and Electronic Engineering
University of Dhaka
Dhaka, Bangladesh

***Abstract*—Designing new storage solutions for optimized performance is an active research area. In this paper, we discuss the emerging storage technologies that show the potential to significantly improve the performance of the applications. Specifically, we summarize the key benefits and applications of such four storage technologies, i.e., Zoned Namespace SSD, Key-value SSD, Multi-Stream SSD, and Non-volatile Main Memory.**

I. Introduction

The increased availability and usage of digital devices are producing an overwhelming amount of data. Nowadays, 2.5 quintillion bytes of data are being generated every day [1]. To store this vast amount of data, efficient storage technologies are essential. Due to this timely demand for optimized storage solutions, the storage research world is actively proposing modern storage solutions. In this paper, we discuss the background and related works on some of these emerging storage technologies.

Hard-Disk-Drives (HDD) have already been replaced the Solid-State-Drives (SSD) because of SSDs' improved performance and enhanced capacity. However, recent works focus on emerging storage technologies that can outperform the traditional SSDs in many cases. We discuss such four storage technologies, i.e., Zoned Namespace SSD, Key-value SSD, Multi-Stream SSD, and Non-volatile Main Memory. We first discuss the design of these techniques, then we focus on the key advantages and limitations of each of these solutions, and finally, we summarize some of the key works on these storage solutions.

II. Emerging Storage Technologies

A. Zoned Namespace SSD

The Zoned Namespace SSD (ZNS) differs from the traditional SSDs by dividing the address space into multiple equally-sized *zones*. Unlike the traditional SSDs, which supports random writes, the *zones* only support append-only, i.e., sequential writes to any zone. Read operation in ZNS remains same. The ZNS interface which replaces the previous block interface on the traditional SSDs, also eliminates the costly garbage collection, media over-provisioning, and reduces DRAM usage. This is because, in ZNS, the control is given to the host to manage the write amplification, and ZNS enables the host to take data placement and I/O scheduling decisions by being more workload aware. The popularity of the ZNS is increasing so rapidly that [2] argues that research efforts on the traditional SSDs is obsolete and the research community should focus on the ZNS instead.

Although ZNS devices have been standardized, there are many unsolved research opportunities, such as the improvement of *zone* management, the interaction between the applications and *zones*, finding the best I/O scheduling for the host, and others [2]. Ref. [3], [4] investigate the impact of different ZNS features on the performance. Ref. [5], [6] analyzes how log-based file systems, databases, and LSM trees can benefit from the ZNS's append-only *zone* structure. LSM-based garbage collection for *zones* is proposed by [7]. The *ZNS+* interface is introduced to further optimize the ZNS interface to support in-storage *zone* compaction [8].

B. Key-value SSD

Key-value SSDs (KV-SSD) are another emerging technology that specifically aims to improve the performance of the key-value stores. In the traditional SSDs, the key-value stores run on the top of the block interface which maintains all the mappings, from the key-values to the flash pages. However, KV-SSDs eliminate all the data management overhead of the block layer and provides direct access to data to the key-value storage applications. Ref. [9] describes the design of the KV-SSDs in detail. Among the recent works, [10] discusses the advantages and limitations of the KV-SSDs. Ref. [11] introduces hybrid data reliability for the key-value devices. *'PinK'* [12] implements LSM-tree-based KV-SSD to improve the tail latency. Others implement concurrency for parallel key-value stores in KV-SSDs [13], enables distributed key-value stores on KV-SSDs [14], [15], new append feature for the KV-SSDs [16], and others [17], [18].

C. Multi-Stream SSD

To improve the endurance and performance of the traditional SSDs, Multi-Stream SSDs(MS-SSD) [19] are introduced [20]. MS-SSDs split the incoming storage data streams into multiple streams and write to physically separated blocks. To split the data stream into multiple streams, MS-SSDs use the lifetime of the data, so that data with similar lifetime could be grouped together for improved lifetime and performance. There are several works on stream management techniques. Ref. [21] introduces fully automatic stream identification, [22], [23] use workload features for stream identification, and [24] analyzes workload in the runtime to find the best stream identification technique. To automatically tune the parameters used by different stream identification techniques, [25] designs a novel framework for the MS-SSDs. Ref. [26] proposes the concept of virtual streams for the MS-SSDs to overcome the limited numbers of streams supported by the device. Other recent works on MS-SSDs include techniques for garbage collection in MS-SSDs [27], [28] and methods for maintaining the quality of service of streams [29].

D. Non-volatile Main Memory

In addition to the emerging SSD technologies discussed above, Non-volatile Main Memory(NVMM) devices are new types of memory and storage solutions that provide some unique properties. NVMMs come in the DIMM form factor and thus can be directly connected to the mainboard. Hence, NVMMs can be used as both storage and memory devices. NVMMs provide lower access latency than the SSDs but higher than the DRAM. To exploit the performance and byte-addressability provided by the NVMMs, there is an extensive amount of work that implements filesystems [30]–[34] and optimized interfaces for NVMMs [35]–[37]. Moreover, some works [38], [39] focus on the data security on the NVMM devices. Furthermore, improved hybrid memory systems have been designed using DRAM and NVMMs. State-of-the-art works in designing hybrid memory systems include [40]–[43].

III. Conclusion

In this work, we summarize and discuss four emerging storage technologies. We expect that the background knowledge and relevant works on these emerging technologies provided in this paper will help the research community to design better storage solutions.

References

[1] D. Price, "Infographic: How much data is produced every day?" Mar. 2015. [Online]. Available: https://cloudtweaks.com/2015/03/how-much-data-is-produced-every-day/

[2] T. Stavrinos, D. S. Berger, E. Katz-Bassett, and W. Lloyd, "Don't be a blockhead: Zoned namespaces make work on conventional ssds obsolete," in *Proceedings of the Workshop on Hot Topics in Operating Systems*, ser. HotOS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 144–151. [Online]. Available: https://doi.org/10.1145/3458336.3465300

[3] H. Shin, M. Oh, G. Choi, and J. Choi, "Exploring performance characteristics of zns ssds: Observation and implication," in *2020 9th Non-Volatile Memory Systems and Applications Symposium (NVMSA)*, 2020, pp. 1–5.

[4] U. Maheshwari, "From blocks to rocks: A natural extension of zoned namespaces," in *Proceedings of the 13th ACM Workshop on Hot Topics in Storage and File Systems*, ser. HotStorage '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 21–27. [Online]. Available: https://doi.org/10.1145/3465332.3470870

[5] D. R. Purandare, P. Wilcox, H. Litz, and S. Finkelstein, "Append is near: Log-based data management on zns ssds," in *12th Annual Conference on Innovative Data Systems Research (CIDR'22).*, 2022.

[6] M. Bjørling, A. Aghayev, H. Holmberg, A. Ramesh, D. L. Moal, G. R. Ganger, and G. Amvrosiadis, "ZNS: Avoiding the block interface tax for flash-based SSDs," in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*. USENIX Association, Jul. 2021, pp. 689–703. [Online]. Available: https://www.usenix.org/conference/atc21/presentation/bjorling

[7] G. Choi, K. Lee, M. Oh, J. Choi, J. Jhin, and Y. Oh, "A new LSM-style garbage collection scheme for ZNS SSDs," in *12th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 20)*. USENIX Association, Jul. 2020. [Online]. Available: https://www.usenix.org/conference/hotstorage20/presentation/choi

[8] K. Han, H. Gwak, D. Shin, and J. Hwang, "Zns+: Advanced zoned namespace interface for supporting in-storage zone compaction," in *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*. USENIX Association, Jul. 2021, pp. 147–162. [Online]. Available: https://www.usenix.org/conference/osdi21/presentation/han

[9] Y. Kang, R. Pitchumani, P. Mishra, Y.-s. Kee, F. Londono, S. Oh, J. Lee, and D. D. G. Lee, "Towards building a high-performance, scale-in key-value storage system," in *Proceedings of the 12th ACM International Conference on Systems and Storage*, ser. SYSTOR '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 144–154. [Online]. Available: https://doi.org/10.1145/3319647.3325831

[10] M. P. Saha, A. Maruf, B. S. Kim, and J. Bhimani, "Kv-ssd: What is it good for?" in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1105–1110.

[11] R. Pitchumani and Y.-S. Kee, "Hybrid data reliability for emerging Key-Value storage devices," in *18th USENIX Conference on File and Storage Technologies (FAST 20)*. Santa Clara, CA: USENIX Association, Feb. 2020, pp. 309–322. [Online]. Available: https://www.usenix.org/conference/fast20/presentation/pitchumani

[12] J. Im, J. Bae, C. Chung, Arvind, and S. Lee, "PinK: High-speed in-storage key-value store with bounded tails," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. USENIX Association, Jul. 2020, pp. 173–187. [Online]. Available: https://www.usenix.org/conference/atc20/presentation/im

[13] J. Bhimani, J. Yang, N. Mi, C. Choi, M. Saha, and A. Maruf, "Fine-grained control of concurrency within kv-ssds," in *Proceedings of the 14th ACM International Conference on Systems and Storage*, ser. SYSTOR '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3456727.3463777

[14] T. Bisson, K. Chen, C. Choi, V. Balakrishnan, and Y.-s. Kee, "Crail-kv: A high-performance distributed key-value store leveraging native kv-ssds over nvme-of," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, 2018, pp. 1–8.

[15] C. Chung, J. Koo, J. Im, Arvind, and S. Lee, "Lightstore: Software-defined network-attached key-value drives," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 939–953. [Online]. Available: https://doi.org/10.1145/3297858.3304022

[16] B. Ganeshan, A. Dininno, and G. G. Basava, "Key value append," Oct. 22 2020, uS Patent App. 16/440,978.

[17] S.-H. Kim, J. Kim, K. Jeong, and J.-S. Kim, "Transaction support using compound commands in {Key-Value}{SSDs}," in *11th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 19)*, 2019.

[18] C. Choi, T. Bisson, and K. Chen, "Native key-value storage enabled distributed storage system," Mar. 29 2022, uS Patent 11,287,994.

[19] J.-U. Kang, J. Hyun, H. Maeng, and S. Cho, "The multi-streamed Solid-State drive," in *6th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 14)*. Philadelphia, PA: USENIX Association, Jun. 2014. [Online]. Available: https://www.usenix.org/conference/hotstorage14/workshop-program/presentation/kang

[20] J. Bhimani, J. Yang, Z. Yang, N. Mi, N. H. V. K. Giri, R. Pandurangan, C. Choi, and V. Balakrishnan, "Enhancing ssds with multi-stream: What? why? how?" in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, 2017, pp. 1–2.

[21] T. Kim, D. Hong, S. S. Hahn, M. Chun, S. Lee, J. Hwang, J. Lee, and J. Kim, "Fully automatic stream management for Multi-Streamed SSDs using program contexts," in *17th USENIX Conference on File and Storage Technologies (FAST 19)*. Boston, MA: USENIX Association, Feb. 2019, pp. 295–308. [Online]. Available: https://www.usenix.org/conference/fast19/presentation/kim-taejin

[22] J. Bhimani, N. Mi, Z. Yang, J. Yang, R. Pandurangan, C. Choi, and V. Balakrishnan, "Fios: Feature based i/o stream identification for improving endurance of multi-stream ssds," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2018, pp. 17–24.

[23] J. Bhimani, Z. Yang, J. Yang, A. Maruf, N. Mi, R. Pandurangan, C. Choi, and V. Balakrishnan, "Automatic stream identification to improve flash endurance in data centers," *ACM Trans. Storage*, vol. 18, no. 2, apr 2022. [Online]. Available: https://doi.org/10.1145/3470007

[24] J. Yang, R. Pandurangan, C. Choi, and V. Balakrishnan, "Autostream: Automatic stream management for multi-streamed ssds," in *Proceedings of the 10th ACM International Systems and Storage Conference*, ser. SYSTOR '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3078468.3078469

[25] J. Bhimani, A. Maruf, N. Mi, R. Pandurangan, and V. Balakrishnan, "Auto-tuning parameters for emerging multi-stream flash-based storage drives through new i/o pattern generations," *IEEE Transactions on Computers*, vol. 71, no. 2, pp. 309–322, 2022.

[26] H. Yong, K. Jeong, J. Lee, and J.-S. Kim, "vStream: Virtual stream management for multi-streamed SSDs," in *10th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 18)*. Boston, MA: USENIX Association, Jul. 2018. [Online]. Available: https://www.usenix.org/conference/hotstorage18/presentation/yong

[27] S. G. Fischer, C. Choi, J. Martineau, and R. Pandurangan, "Methods for multi-stream garbage collection," Jun. 29 2021, uS Patent 11,048,624.

[28] J. Martineau and C. Choi, "Enhanced multi-stream operations," Jan. 30 2018, uS Patent 9,880,780.

[29] H. Hingkwan, C. Choi, D. Tseng, and J. Huo, "Multi-stream ssd qos management," Mar. 17 2020, uS Patent 10,592,171.

[30] J. Condit, E. B. Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, and D. Coetzee, "Better i/o through byte-addressable, persistent memory," in *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles*, ser. SOSP '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 133–146. [Online]. Available: https://doi.org/10.1145/1629575.1629589

[31] S. R. Dulloor, S. Kumar, A. Keshavamurthy, P. Lantz, D. Reddy, R. Sankaran, and J. Jackson, "System software for persistent memory," in *Proceedings of the Ninth European Conference on Computer Systems*, ser. EuroSys '14. New York, NY, USA: Association for Computing Machinery, 2014. [Online]. Available: https://doi.org/10.1145/2592798.2592814

[32] Y. Chen, J. Shu, J. Ou, and Y. Lu, "Hinfs: A persistent memory file system with both buffering and direct-access," *ACM Transactions on Storage (ToS)*, vol. 14, no. 1, pp. 1–30, 2018.

[33] J. Yang, J. Izraelevitz, and S. Swanson, "Orion: A distributed file system for {Non-Volatile} main memory and {RDMA-Capable} networks," in *17th USENIX Conference on File and Storage Technologies (FAST 19)*, 2019, pp. 221–234.

[34] S. Zheng, M. Hoseinzadeh, and S. Swanson, "Ziggurat: A tiered file system for {Non-Volatile} main memories and disks," in *17th USENIX Conference on File and Storage Technologies (FAST 19)*, 2019, pp. 207–219.

[35] M. Honda, G. Lettieri, L. Eggert, and D. Santry, "{PASTE}: A network programming interface for {Non-Volatile} main memory," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018, pp. 17–33.

[36] J. Guerra, L. Marmol, D. Campello, C. Crespo, R. Rangaswami, and J. Wei, "Software persistent memory," in *2012 USENIX Annual Technical Conference (USENIX ATC 12)*. Boston, MA: USENIX Association, Jun. 2012, pp. 319–331. [Online]. Available: https://www.usenix.org/conference/atc12/technical-sessions/presentation/guerra

[37] H. Volos, A. J. Tack, and M. M. Swift, "Mnemosyne: Lightweight persistent memory," in *Proceedings of the Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS XVI. New York, NY, USA: Association for Computing Machinery, 2011, p. 91–104. [Online]. Available: https://doi.org/10.1145/1950365.1950379

[38] P. Zuo and Y. Hua, "{SecPM}: a secure and persistent memory system for non-volatile memory," in *10th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 18)*, 2018.

[39] S. Mittal and A. I. Alsalibi, "A survey of techniques for improving security of non-volatile memories," *Journal of Hardware and Systems Security*, vol. 2, no. 2, pp. 179–200, 2018.

[40] Z. Yan, D. Lustig, D. Nellans, and A. Bhattacharjee, "Nimble page management for tiered memory systems," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019, pp. 331–345.

[41] J. Kim, W. Choe, and J. Ahn, "Exploring the design space of page management for Multi-Tiered memory systems," in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*. USENIX Association, Jul. 2021, pp. 715–728. [Online]. Available: https://www.usenix.org/conference/atc21/presentation/kim-jonghyeon

[42] A. Maruf, A. Ghosh, J. Bhimani, D. Campello, A. Rudoff, and R. Rangaswami, "Multi-clock: Dynamic tiering for hybrid memory systems," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2022.

[43] A. Maruf, J. Bhimani, A. Ghosh, and R. Rangaswami, "Machine learning based tiered memory systems and methods," Apr. 19 2022, uS Patent 11,307,986.