OC-DLRM: Minimizing the I/O Traffic of DLRM between Main Memory and OCSSD

Shang-Hung Ti[1], Tseng-Yi Chen[2], Tsung Tai Yeh[3], Shuo-Han Chen[4], Yu-Pei Liang[5]
[1,2]*Department of Computer Science and Information Engineering, National Central University*, Taoyuan, Taiwan.
[3]*Department of Computer Science, National Yang Ming Chiao Tung University*, Hsinchu, Taiwan.
[4]*Institute of Artificial Intelligence Innovation, National Yang Ming Chiao Tung University*, Hsinchu, Taiwan.
[5]*Department of Computer Science and Information Engineering, National Chung Cheng University*, Chiayi, Taiwan.

***Abstract*—Due to the exponential growth of data in computing, DRAM-based main memory is now insufficient for data-intensive applications like machine learning and recommendation systems. This has led to a performance issue involving data transfer between main memory and storage devices. Conventional NAND-based SSDs are unable to efficiently handle this problem as they can't distinguish between data types from the host system. In contrast, open-channel SSDs (OCSSD) offer a solution by optimizing data placement from the host-side system. This research focuses on developing a new data access model for deep learning recommendation systems (DLRM) using OCSSD storage drives, called OC-DLRM. OC-DLRM reduces I/O traffic to flash memory by aggregating frequently-accessed data using the I/O unit of a flash memory drive. Our experiments show that OC-DLRM has significant performance improvement compared with traditional swapping space management techniques.**

***Index Terms*—Open channel SSD, flash memory, deep learning, recommendation model, virtual memory**

I. Background & Motivation

A. Deep Learning Recommendation Model (DLRM)

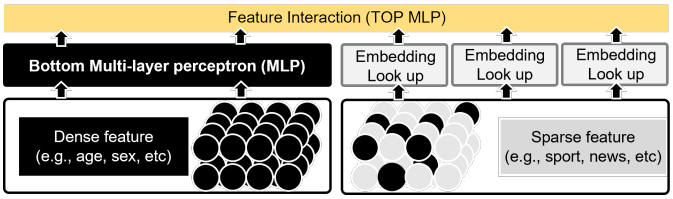

Fig. 1. The process of deep learning recommendation model.

Recommendation algorithms are crucial in various domains such as social networks, e-commerce, and video streaming. Deep learning, particularly Facebook's Deep Learning Recommendation Model (DLRM), has greatly improved recommendation accuracy by capturing complex user preferences. This integration of deep learning has transformed recommendations, enhancing user experiences and business outcomes. In Figure 1, we illustrate the DLRM process. Input data is divided into dense (e.g., age, sex, address) and sparse (e.g., news, advertisement, sport scores) features. Dense features go through a bottom multi-layer perceptron (MLP), followed by a top MLP. Sparse features are processed using embeddings, where multiple tables represent them, and an embedding lookup method selects vectors. These embeddings, along with dense features, input into the top MLP, training the recommendation model. Optimizing storage throughput is essential for speeding up DLRM's training, as embedding tables are typically large (gigabyte-sized) and stored on a storage device.

B. Open Channel SSD (OCSSD)

Solid-state drives (SSDs) are widely used in data centers due to their high performance and decreasing costs. An SSD comprises multiple chips, each containing numerous planes, and each plane has thousands of flash blocks, each with hundreds of flash pages. Pages are for read/write, and blocks are for erasure. SSDs offer high parallelism as each plane can be accessed independently.

NAND flash memory cells are the basic storage units in SSDs, differing significantly from traditional hard disk drives (HDDs). To abstract these differences for system developers, SSDs integrate a flash translation layer (FTL) management. FTL consists of three key components: mapping structure [3], [4], a garbage collection mechanism [5], [6], and a wear-leveling strategy [7], [8]. The mapping structure records logical-to-physical block addresses, enabling out-place-updates. Garbage collection reclaims space, and wear-leveling maximizes flash block lifetimes. However, FTL management within SSDs limits host system optimization of SSD performance.

In recent years, the pursuit of enhanced SSD performance has led to innovative architectures like open-channel SSDs (OCSSD) [2]. OCSSD transfers FTL management from the SSD to the host system, exposing the physical layout of flash memory spaces. This paradigm shift allows the host system to optimize data placement and drive garbage collection, offering unprecedented control.

With OCSSD, the host system intelligently places data across physical flash memory addresses, maximizing efficiency. All FTL components, including address mapping and wear-leveling algorithms, are moved to the host system. Hardware engines like ECC and encryption/decryption are retained in the flash memory device for data integrity and security.

OCSSD's primary goal is to empower the host system with control over data placement strategies, elevating storage performance. Researchers and practitioners can leverage OCSSD

979-8-3503-4859-0/DATE24/© 2024 EDAA

to explore new avenues for storage technology advancement and optimization of host-system interactions.

II. OC-DLRM Scheme

Our study aims to minimize data traffic between memory and storage space. We propose an OCSSD-friendly DLRM memory management scheme, OC-DLRM, to place frequently-updated embedding vectors in the same flash page and evenly distribute access operations across channels for OCSSD's high parallelism. Figure 2 shows OC-DLRM's system architecture. OC-DLRM comprises three main components: embedding cache management, two-level mapping (chunk-level for embedding tables and vector-level for updated vectors), and space reclamation. The chunk-level mapping leverages OCSSD's parallelism, while vector-level mapping aggregates updated vectors within the same flash page to reduce data movement between memory and storage. The embedding cache management handles vector eviction from memory to storage. To save host memory space used by the vector-level mapping, OC-DLRM integrates space reclamation, recycling memory by merging updated vectors with the original chunk data.

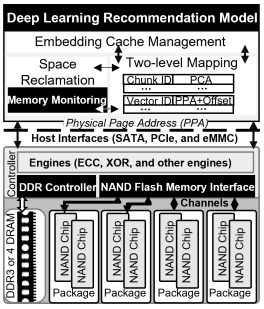

Fig. 2. The system architecture of OC-DLRM.

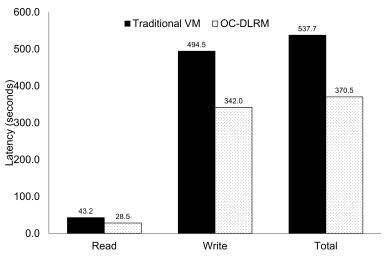

Fig. 3. The latency in RMC 1.

III. Experiments

A. Environmental Settings

In this section, we comprehensively evaluate OC-DLRM by analyzing key performance metrics, including data movement, latency, and throughput. We integrated OC-DLRM into an open-source OCSSD simulator [9] within the QEMU environment to ensure the validity and reliability of our experimental findings, closely emulating real-world OCSSD systems. To incorporate virtual memory management into the DLRM framework, we used LibTorch [10], a C++ library, balancing ease of implementation with robustness. We meticulously configured the OCSSD device for experiments, summarizing critical parameters in Table I, covering aspects like memory capacity, data placement policies, and flash memory characteristics. This ensures a comprehensive exploration of OC-DLRM's performance in diverse scenarios, reflecting real-world OCSSD deployments. To assess OC-DLRM's impact on different training data patterns, we used three built-in models

TABLE I
Configuration for OCSSD [2].

OCSSD Configuration			
Channels	8	# of Planes	4
PUs per channel	4	Page Size	16KB
Sector size	4 KBs	Block Size	256 Pages
Performance (Single PU in Sequential Mode)			
Write	47MB/s	Read	280MB/s
MAX. Write	4GB/s	MAX. Read	4.5GB/s

RMC 1 from the DLRM project [1], representing various intensity levels,.

According to Figure 3, we can observe that read latency is much smaller than write latency in both solutions because RMC 1 model contains a small number of embedding tables but a high frequency of looking up embedding tables. Compared with the baseline, our OC-DLRM can reduce the latency by 31.0%.

IV. Concluding Remarks

This study aims to enhance DLRM's performance, especially in the embedding process, which is memory capacity-dependent. Existing research has proposed storage accelerators for DLRM but often involve extra hardware costs or focus on the inference stage. There is currently no research on leveraging OCSSD to optimize DLRM's training. To tackle this issue, we present OC-DLRM, a novel framework that reduces data movement between memory and storage spaces while improving DLRM's training performance by capitalizing on OCSSD's high parallelism.

References

[1] M. Naumov and D. Mudigere, "Deep learning recommendation model for personalization and recommendation systems.", [online] Available: https://ai.facebook.com/blog/dlrm-an-advanced-open-source-deep-learning-recommendation-model/, 2019.

[2] M. Bjørling, et al., "LightNVM: The Linux Open-Channel SSD Subsystem.", 15th USENIX Conference on File and Storage Technologies (FAST 17), Feb. 2017.

[3] S. Chen, et al., "Beyond Address Mapping: A User-Oriented Multi-Regional Space Management Design for 3D NAND Flash Memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 39, no. 6, pp. 1286-1299, Jun. 2020.

[4] M. Yang, et al., "Capacity-independent Address Mapping for Flash Storage Devices with Explosively Growing Capacity," IEEE Transactions on Computers (TC), vol. 65, no. 2, pp. 448-465, Feb. 2016.

[5] T. Chen, et al., "VirtualGC: Enabling Erase-free Garbage Collection to Upgrade the Performance of Rewritable SLC NAND Flash Memory," ACM/IEEE Design Automation Conference (DAC), Austin, Texas, USA, Jun. 18-22, 2017.

[6] J. Kim, et al., "Alleviating Garbage Collection Interference Through Spatial Separation in All Flash Arrays," 2019 USENIX Annual Technical Conference (USENIX ATC 19), pp. 799–812, 2019.

[7] Y. Chang, et al., "Improving Flash Wear-Leveling by Proactively Moving Static Data," IEEE Transactions on Computers (TC), vol. 59, no. 1, pp. 53-65, Jan. 2010.

[8] Dharamjeet, et al., "LLSM: A Lifetime-Aware Wear-Leveling for LSM-Tree on NAND Flash Memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 41, no. 11, pp. 3946-3956, Nov. 2022.

[9] LightNVM, "Open-Channel SSD Simulator", http://lightnvm.io/pblk-tools/ocssd.html

[10] The Linux Foundation, "LibTorch", https://pytorch.org/cppdocs/installing.html