RecSSD: Near Data Processing for Solid State Drive Based Recommendation Inference Extended Abstract

Mark Wilkening[1], Udit Gupta[1,2], Samuel Hsia[1], Caroline Trippel[2], Carole-Jean Wu[2], David Brooks[1], and Gu-Yeon Wei[1]

[1]Department of Computer Science, Harvard University
[2]Facebook

1. Motivation

Recommendation algorithms are used across a variety of Internet services such as social media, entertainment, e-commerce, and search [9, 18, 6, 20, 19, 15]. Deep learning based recommendation models consume a significant portion of datacenter capacity and compute cycles. Compared to other AI-driven applications, recommendation accounts for 10× the capacity in Facebook's datacenter [9, 13]. Similar capacity requirements can be found across Google, Alibaba, Baidu, and Amazon [18, 20, 19].

One of the key distinguishing features of recommendation models is processing categorical input features using embedding tables. Categorical inputs are processed in two steps: (1) multi-hot encoded vectors are used to *gather* specific rows within an embedding table, and (2) the indexed rows produce embedding vectors which are *aggregated*, subsequently. These embedding table operations transform large, highly sparse categorical input features into low-dimensional vectors.

Larger embeddings better capture the distribution of categorical features. For instance, in Baidu's recommendation system, an order of magnitude increase in the embedding table size translates into a significant 2% accuracy gain [4]. Because of these accuracy incentives, the size of state-of-the-art deep learning recommendation models has grown dramatically, from a few tens of GBs to the TB region [12, 9, 14, 16, 17, 18]. In fact, in many cases, the size of recommendation models is set by the amount of memory available on servers. Unfortunately, fast and high-capacity DDR-based main memory incurs heavy infrastructure cost in datacenters [3].

A promising alternative is to store embedding tables in SSDs. While offering orders of magnitude higher storage density than main memory, SSDs exhibit slower read and write performance. The read bandwidth of SSD is on the order of 1-3 GB/s, 25× worse than DRAM (e.g., 75 GB/s). Though SSDs can accommodate the growing capacity demand of recommendation, the much lower bandwidth and longer latency as compared to DRAM main memory systems threaten the viability of SSDs. This paper proposes RecSSD to tackle these challenges and provide high capacity storage while maintaining performance, with specialized Near-Data-Processing (NDP) hardware to optimize SSD performance.

2. Limitations of the State of the Art

Given the growing importance of recommendation models, researchers have started proposing specialized systems and hardware solutions to improve the execution efficiency [9, 8, 10, 11]. In particular, to address the irregular memory access patterns and the model-specific memory bandwidth requirement of recommendation models, recent work accelerates low-compute intensity embedding operations using near-memory processing hardware [10, 11]. However, one of the important challenges to enabling efficient datacenter scale deployment to real-time recommendation serving is to address the growing model capacity demand. In particular, the ever-increasing embedding table size, from GB to TB scale, pushes the limits of previously-studied DRAM systems. Anticipating even larger model capacity, one promising alternative to DRAM-based main memory systems is to use SSD-based storage systems. There are a few recent prior work examining the role of SSDs for industry-scale neural recommendation execution [16, 17]. However, the focus of prior work is on enabling and optimizing the training of industry-scale neural recommendation that cannot be applied directly for inference optimization.

In addition to training, recent work explores SSDs for recommendation inference. For instance, Bandana simulates the presence of SSD systems for Facebook's production recommendation inference systems [7]. To enable high-capacity and high-performance recommendation systems, Bandana caches frequently accessed embedding vectors in the DRAM memory. Bandana, however, is limited to a particular subset of Facebook's recommendation models. In contrast, RecSSD proposes an orthogonal set of optimization techniques by directly improving SSD performance using NDP. Furthermore, we evaluate RecSSD over eight different, industry-representative recommendation models on a real OpenSSD system.

3. Key Insights

System design and optimization for deep learning-based ranking and recommendation use cases is at a nascent stage, despite the exponentially-increasing importance. To leverage the capacity advantage of solid state drives (SSDs), RecSSD stores TB-scale embedding tables in the SSD (instead of DRAM). We recognize the viability of SSD storage for use in recom-

mendation inference across a variety of industry-representative workloads using a real system evaluation. Additionally, we identify embedding-bound models which struggle to effectively utilize SSD storage because of the reduced latency and bandwidth performance relative to main memory systems.

Given these performance challenges, this is the first paper that identifies and evaluates the opportunities of utilizing SSD-based near data processing (NDP) solutions tailor-designed for neural recommendation inference. To mitigate the increasing data movement latency and energy overhead, we propose offloading the simple *gather-aggregate* computation of the embedding operator to computational resources within the SSD system. By doing so, RecSSD fully utilizes the internal SSD bandwidth, leading to significantly reduced communication between the CPU host and the SSDs.

Furthermore, embedding vector access patterns expose cache optimization opportunities. We provide a detailed reuse characterization based on industry-scale recommendation inference serving logs, in order to inform our locality-based optimizations. Importantly however, the variation in the observed reuse patterns suggests that, although in some cases, caching can be used to effectively deal with block access, strategies for efficiently handling sparse accesses are also needed. The NDP techniques proposed in RecSSDcomplement the previous caching techniques but focus on improving the operation latency, which must bypass host caching techniques and access SSD storage. We evaluate the performance of RecSSD alongside caching optimizations across a range of locality conditions informed by the memory locality characterization (Section 3).

4. Main Artifacts

In this paper we present RecSSD, a *near-data processing* (NDP) solution for efficient recommendation embedding table execution on SSD memory. First, compared to baseline storage systems, RecSSD increases the effective bandwidth to Flash memories by utilizing the internal SSD bandwidth rather than using PCIe. Next, RecSSD reduces command and control overheads in the host driver stack by reducing the number of I/O commands needed. Finally, RecSSD greatly reduces the amount of data transmitted over PCIe by packing output results into logical blocks. To highlight the efficacy of the proposed solution, we implement RecSSD within the FTL of a real SSD system and evaluate the solution over eight industry-representative recommendation models. Here we detail how RecSSD is implemented and evaluated.

OpenSSD and Micron UNVMe We implement a fully-functional NDP SLS operator in the open source Cosmos+ OpenSSD system [2]. In order to provide a feasible solution for datacenter deployment, we implement RecSSD within the FTL firmware requiring no hardware changes. To maintain compatibility with the NVMe protocols, the RecSSD interface is implemented within Micron's UNVMe driver library [5] and enables flexible input data and command configurations.

Neural recommendation models To evaluate RecSSD, we use a diverse set of eight industry-representative recommendation models provided in DeepRecInfra [8], implemented in Python using Caffe2 [1]. In order to evaluate the performance of end-to-end recommendation models on real systems, we integrate the SparseLengthsSum operations (embedding table operations in Caffe2) with our custom-designed NDP solution.

Input traces In addition to the models themselves, we instrument the open-source synthetic trace generators from Facebook's open-sourced DLRM [13] with our locality analysis from production-scale recommendation systems.

5. Key Results and Contributions

The key results and contributions of this work are:

- We design RecSSD, the first NDP-based SSD system for recommendation inference. Improving the performance of conventional SSD systems, the proposed design targets the main performance bottleneck to datacenter scale recommendation execution using SSDs. Furthermore, the latency improvement further enables recommendation models with higher storage capacities at reduced infrastructure cost.
- We implement RecSSD in a real system on top of the Cosmos+OpenSSD hardware. The implementation demonstrates the viability of Flash-based SSD systems for industry-scale recommendation model execution. In order to provide a feasible solution for datacenter scale deployment, we implement RecSSD within the FTL firmware; the interface is compatible with existing NVMe protocols, requiring no hardware changes.
- We evaluate the proposed design across eight industry-representative models across various use cases (e.g., social media, e-commerce, entertainment, search). Of the eight, our real system evaluation shows that five models — whose runtime is dominated by compute-intensive FC layers — achieve comparable performance on the SSD system compared to the DRAM baseline. The remaining three models are dominated by memory-bound, embedding table operations. On top of the highly optimized hybrid DRAM-SSD systems, we demonstrate that RecSSD improves performance by up to 4× for individual embedding operations, translating into up to 2× end-to-end recommendation inference latency reduction.

6. Why ASPLOS

Following ASPLOS' call for multi-disciplinary systems research, RecSSD uses vertically-integrated research methodologies to optimize recommendation systems. The work crosses the intersection of machine learning, operating systems, memory systems, and hardware. We evaluate RecSSD on real hardware systems to demonstrate the efficacy of the proposed NDP hardware, while providing detailed performance analysis to foster intuition on future research in NDP and SSD based recommendation.

References

[1] Caffe2.

[2] Cosmos+ openssd platform.

[3] Dram prices continue to climb.

[4] Training massive scale deep learning ads systems with gpus and ssds.

[5] Unvme - a user space nvme driver project. https://github.com/zenglg/unvme.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10. ACM, 2016.

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

[8] Udit Gupta, Samuel Hsia, Vikram Saraph, Xiu Qiao Wang, Brandon Reagen, Gu-Yeon Wei, Hsien-Hsin S. Lee, David M. Brooks, and Carole-Jean Wu. Deeprecsys: A system for optimizing end-to-end at-scale neural recommendation inference. *ArXiv*, abs/2001.02772, 2020.

[9] Udit Gupta, Xiaodong Wang, Maxim Naumov, Carole-Jean Wu, Brandon Reagen, David Brooks, Bradford Cottel, Kim Hazelwood, Bill Jia, Hsien-Hsin S Lee, et al. The architectural implications of facebook's dnn-based personalized recommendation. *arXiv preprint arXiv:1906.03109*, 2019.

[10] Liu Ke, Udit Gupta, Carole-Jean Wu, Benjamin Y. Cho, Mark Hempstead, Brandon Reagen, Xuan Zhang, David M. Brooks, Vikas Chandra, Utku Diril, Amin Firoozshahian, Kim Hazelwood, Bill Jia, Hsien-Hsin S. Lee, Mengxing Li, Bert Maher, Dheevatsa Mudigere, Maxim Naumov, Martin Schatz, Mikhail Smelyanskiy, and Xiu Qiao Wang. Recnmp: Accelerating personalized recommendation with near-memory processing. *ArXiv*, abs/1912.12953, 2019.

[11] Youngeun Kwon, Yunjae Lee, and Minsoo Rhu. Tensordimm: A practical near-memory processing architecture for embeddings and tensor operations in deep learning. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 740–753, 2019.

[12] Michael Lui, Yavuz Yetim, Özgür Özkan, Zhuoran Zhao, Shin-Yeh Tsai, Carole-Jean Wu, and Mark Hempstead. Understanding capacity-driven scale-out neural recommendation inference, 2020.

[13] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. Deep learning recommendation model for personalization and recommendation systems. *CoRR*, abs/1906.00091, 2019.

[14] Jongsoo Park, Maxim Naumov, Protonu Basu, Summer Deng, Aravind Kalaiah, Daya Khudia, James Law, Parth Malani, Andrey Malevich, Satish Nadathur, et al. Deep learning inference in facebook data centers: Characterization, performance optimizations and hardware implications. *arXiv preprint arXiv:1811.09886*, 2018.

[15] Corinna Underwood. Use cases of recommendation systems in business – current applications and methods, 2019.

[16] Weijie Zhao, Deping Xie, Ronglai Jia, Yulei Qian, Ruiquan Ding, Ming ming Sun, and Ping Li. Distributed hierarchical gpu parameter server for massive scale deep learning ads systems. *ArXiv*, abs/2003.05622, 2020.

[17] Weijie Zhao, Jingyuan Zhang, Deping Xie, Yulei Qian, Ronglai Jia, and Ping Li. Aibox: Ctr prediction model training on a single node. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 319–328, New York, NY, USA, 2019. Association for Computing Machinery.

[18] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. Recommending what video to watch next: A multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 43–51, New York, NY, USA, 2019. ACM.

[19] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5941–5948, 2019.

[20] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068. ACM, 2018.