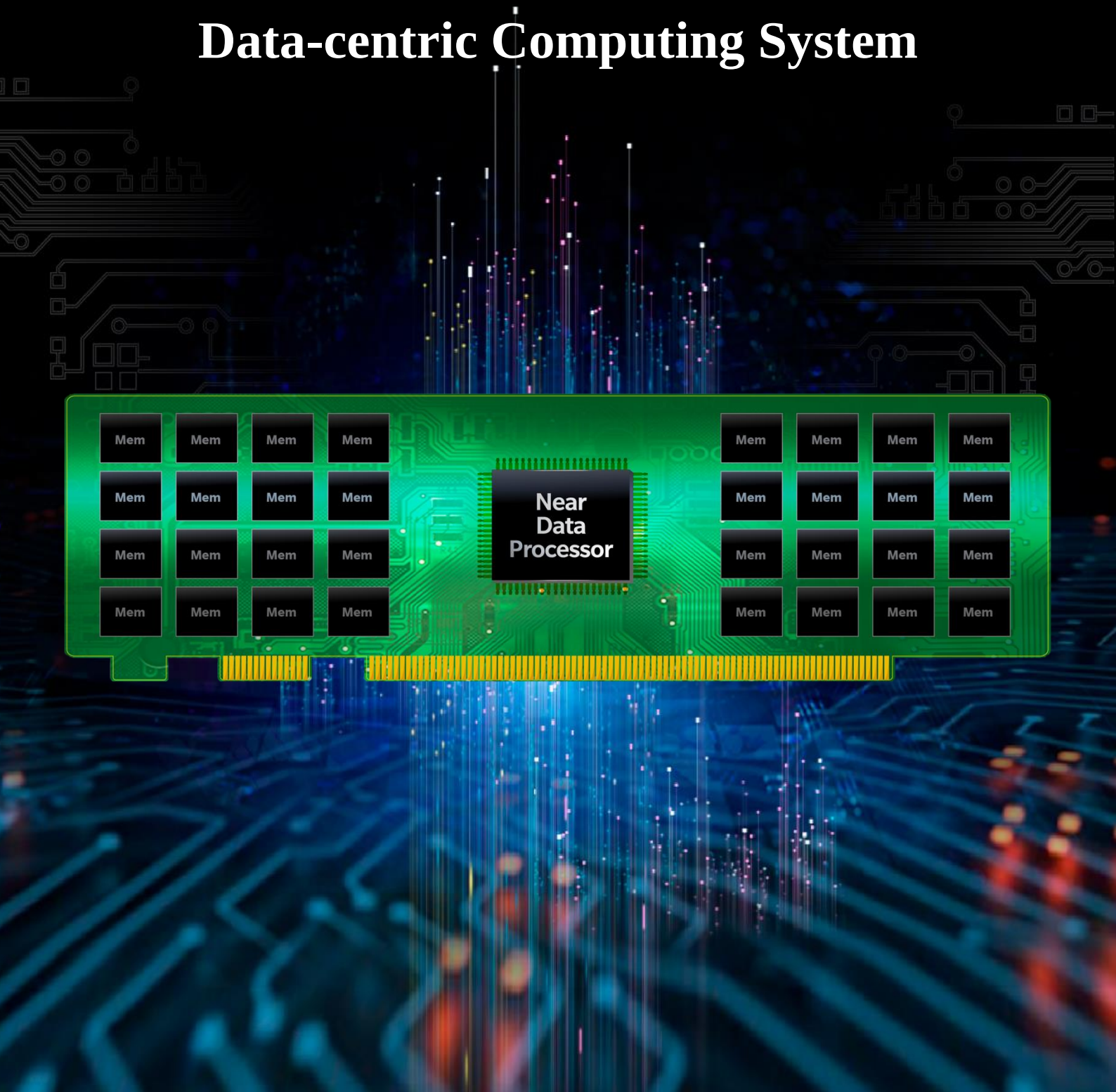
SK hynix

MemAEST

Whitepaper

Computational Memory Solution for Data-centric Computing System



The era of big data and AI is leading to a dramatic increase in data to be processed in systems. Such explosive data growth calls for substantially larger memory capacity and bandwidth, shifting the system performance bottleneck from computing to memory. Many studies seek to address this "memory wall" by offloading applicable functions to the near-memory side. However, most of them overlook the limited capacity of the memory node, which makes communication overhead a critical issue for overall system performance. In this whitepaper, we propose a card-type memory solution that provides extreme-scale memory capacity and bandwidth with a lightweight near data processor (NDP). The NDP can effectively handle data-centric workloads using sufficient internal memory while minimizing data movement to the host. We built our proof-of-concept (PoC) system using FPGAs, and conducted an experimental evaluation with a recommendation system that demonstrated approximately 2.58x performance gains with four such FPGA cards compared to the host.

Introduction

As the demand for big data and AI increases, data is growing explosively in both volume and variety. Such has led to the emergence of data-centric workloads that manipulate and analyze massive amounts of data [1]. Consequently, the burden of data processing and energy consumption from data movement is becoming a critical issue for this rapidly growing segment [2]. For the latest AI models (e.g. Google AI switch [3], Open AI GPT3 [4]) that require large-scale parameters, the energy cost of data movement is substantially higher than that of computation [5]. In some popular Google applications, 62.7% of the total system energy is spent on data movement between CPU and main memory [6]. This energy consumption due to data movement is expected to widen as the era of AI-based big data processing accelerates [7], rendering it essential to reduce data movement in order to improve performance and energy efficiency in data-centric computing systems.

The shift from a compute-centric to a data-driven era is an opportunity for SK hynix to take on a central role in the new ICT (Information & Communications Technology) industry. Having defined a more granular hierarchy for memory in each data processing stage, we are working to make servers and other systems more efficient with targeted solutions such as High-Bandwidth Memory (HBM), the multiprocessor-compatible Compute Express Link (CXL) interface, Processor-in-Memory (PIM) and Computational Memory Solution (CMS). Memory Forest (Mem4EST) shown in Figure 1 is our new initiative and slogan that encapsulates our strategy to build a memory-driven ecosystem with such technical expertise. Just like the lush, green forest it represents, the initiative will generate value from new memory systems and technologies to nurture a wider global ecosystem that produces ESG values for our customers and partners – essentially with Memory for the Environment (E), Society (S), and Tomorrow (T). This paper describes the CMS, one of SK hynix Mem4EST initiatives.

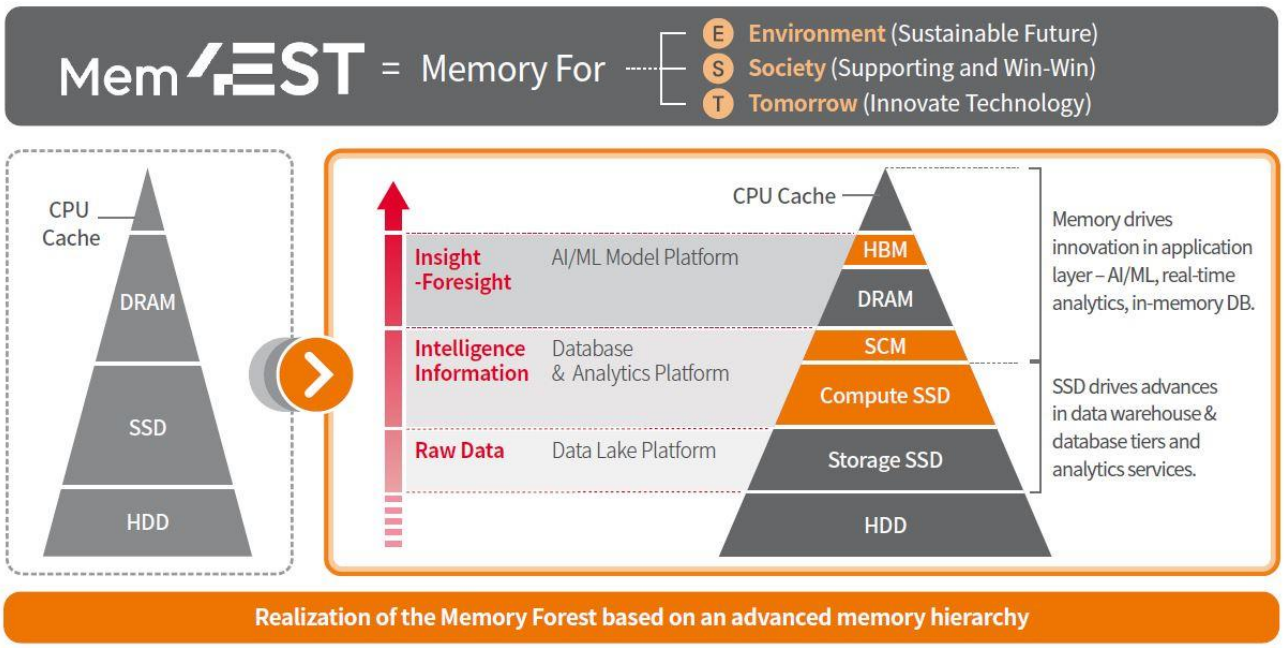


Figure 1. SK hynix Mem4EST Initiatives

Many researchers consider a departure from traditional CPU-centric computing systems, aka Von Neumann architecture, which involves complete separation of the computing and memory units. The work in [8-10] adds extra computing units close to the memory to process the data locally. Processing in memory (PIM) is one of the solution that addresses the data movement issue by processing certain tasks inside memory blocks, resulting in improvements for both performance and energy efficiency [11-14]. However, for some data-intensive workloads, a solution that can reduce inter-node communication by providing sufficient memory capacity and bandwidth to the processing unit is more suitable. In this research, we studied the architecture and use cases for the solution and implemented an FPGA PoC.

Proposed Solution

We propose a novel computational memory solution (CMS) that features a cost-effective, capacity-scalable memory architecture. The proposed solution provides scalable card-type memory expansion which supports large capacity and bandwidth that is additional to the host memory. By placing a lightweight NDP in the memory card, our design can fully utilize extremely high internal memory bandwidth while compensating for the limited bandwidth between the host and expanded memory cards. As the NDP processes data inside the memory card and minimizes data transfer to the host, it can greatly improve the power efficiency of the entire system, also resulting in total cost of ownership (TCO) savings.

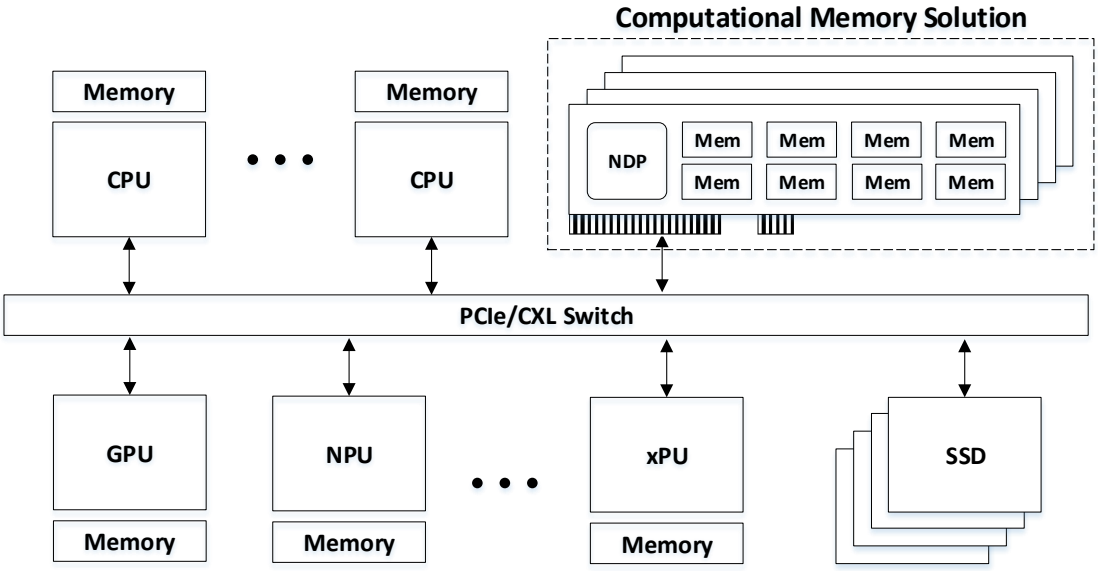


Figure 2. Heterogeneous Computing System Architecture with Computational Memory Solution

Figure 2 shows the heterogeneous computing system architecture with our solution. Here, the CMS specifically handles data-intensive workloads, while accelerators such as GPUs and NPUs handle compute-intensive workloads among various applications. A card-type CMS is an expanded memory node separated from the host memory and connected via the PCIe/CXL interconnect. It consists of multiple modules that provide large-capacity, high-bandwidth memory. A lightweight NDP core is mounted on each module and performs adequate tasks from the data-centric workloads in a parallel manner.

The key benefit of NDP is system performance improvement at reduced power consumption from data movement, enabled by the offloading of tasks to the near-memory side that significantly weigh on data transactions between the memory and host. To fully leverage the system performance gains with NDP, it is crucial to offload the appropriate functions – ones that require intensive data access rather than computing performance [15]. We apply a roofline [16] analysis to identify the data-intensive workloads across applications. The roofline analysis defines the maximum performance that can be obtained with a given system, and thus identifies which operations are suitable for each computing unit. With the analysis, we can extract data-intensive workloads that have low operational intensity (OI) and relatively less computing power requirement from the target application. Then, by providing extremely high memory capacity and bandwidth solutions including an NDP, the proposed design can fully utilize internal bandwidth to accelerate such data-intensive workloads. Figure 3 plots a roofline graph comparing an Intel Xeon server to CMS, footnoted with product-level specifications. It can be seen that workloads with data-intensive characteristics of OI less than 1.6 can run about 5.3x faster in CMS compared to an Intel x86. In addition, the data movement to the host can be minimized by offloading functions, which significantly reduces the result data in the process and thus greatly lowers system power consumption. With the proposed NDP-enabled memory cards, our solution provides cost-effective scalability for the system to scale up and out more easily, without having to pay for expensive servers just to increase the number of memory channels.

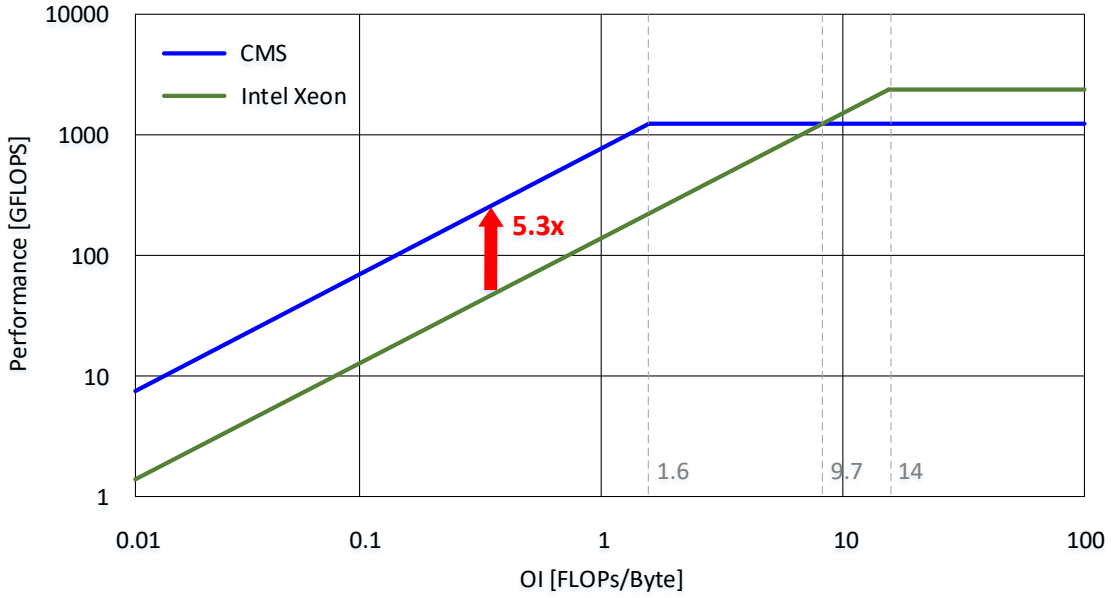


Figure 3. Roofline Analysis

Implementation

We implemented our PoC described heretofore using an FPGA. Figure 4 depicts the HW block diagram of the FPGA PoC. An Intel Xeon server is used as the host, while the CMS PoC is implemented with a Zynq UltraScale+ Sidewinder-100 FPGA board connected to the host server with PCIe interconnect, which will be replaced with CXL IP for memory semantic operations in the near future. ARM Cortex-a53 quad-core is used as the NDP in the FPGA. In this work, we evaluated our design with the memory capacity and bandwidth scaled down to a 16GB and 1-channel DDR4 configuration, but our product roadmap includes an ASIC designed to have hundreds of GBs in capacity and hundreds of GB/s in bandwidth per card with scalability. The mailbox and interrupt controller are responsible for communication between host and NDP, while the DMA handles data transactions. The buffer stores host commands for offloading and the NDP's operation results.

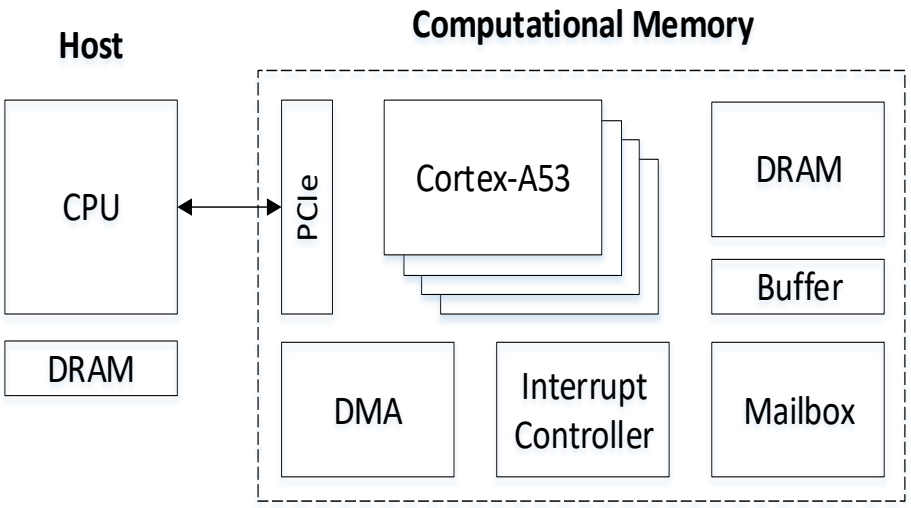


Figure 4. HW Block Diagram of FPGA PoC

The communication flow between the host and NDP is shown in Figure 5. When the host has an offloading workload, it continuously transmits commands to the NDP input buffer using DMA. To notify NDP that a command has been sent, a message loaded with metadata such as the location of the command is written to the mailbox. When the message arrives in the mailbox, an interrupt is generated to the NDP by the interrupt controller. When the interrupt is received, NDP reads the message and performs the job based on the command. After that it writes the result data to the output buffer and delivers a message about the result data to the mailbox. In a similar fashion as before, the interrupt controller sends an interrupt to the host, and the host reads the corresponding message and also reads the result data of NDP using DMA.

The host is connected to multiple NDPs, and communication between them operates in a non-blocking mechanism. That is, as long as each NDP has enough space to receive commands, the host can send a command to multiple NDPs without waiting for the result of the previous request whenever there is an offloading workload. On their part, NDPs can continue to execute their job until the input buffer is completely empty.

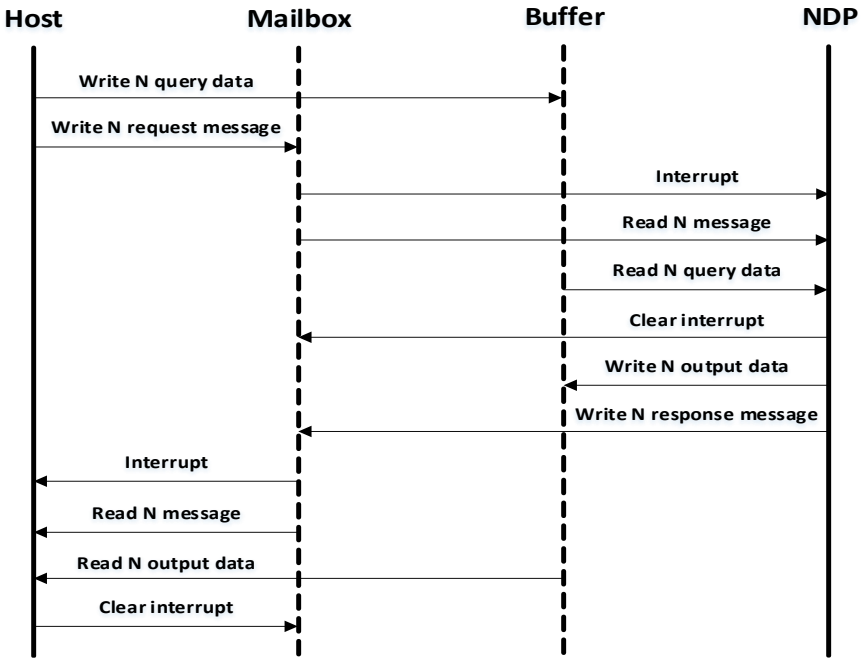


Figure 5. Communication Flow between Host and NDP

Evaluation

To evaluate our implemented CMS system, we selected recommendation as the suitable target application. Recommendation systems are widely used in Internet services such as online shopping, social network, search engine, and video streaming.

Facebook recently proposed "Zion" [17, 18], a unified platform for AI training. Zion can handle diverse deep learning (DL) workloads with complex neural networks similar to the DL-based recommendation system shown in Figure 6, as well as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [19]. However, earlier work has identified technical hurdles to processing the embedding layer [20-22], which has completely different computational characteristics from conventional CNN and RNN workloads, as the embedding layer involves an embedding lookup that consumes a large part of the execution time for both training and inference [20]. This performance bottleneck has been referred to as a "memory wall" to reflect memory capacity and bandwidth challenges [23].

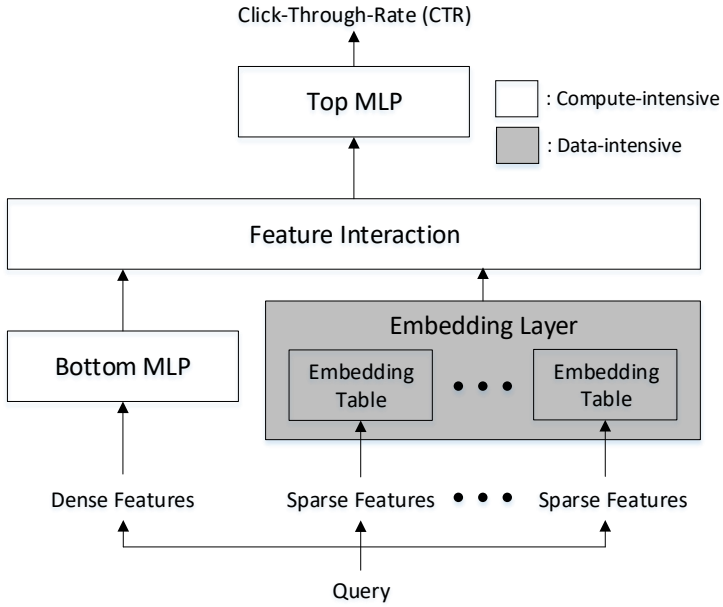


Figure 6. Simplified Scheme for The Modern DL-based Recommendation System

The recommendation system involves the following sequence: First, given features are projected into embedding vectors with the same dimension size. The MLP layer processes dense features (e.g., age, time), while the embedding layer handles sparser features (e.g., gender, item color.) Once each feature is represented by an embedding vector, the feature interaction layer computes the interaction between pairs of features. The upper MLP layer uses these outputs to predict click-through-rate (CTR).

Layer	Inference	Training
Top MLP	7.67	123.57
Feature Interaction	7.87	11.81
Embedding	0.23	1.70
Bottom MLP	6.64	48.01

Table 1. OI of Recommendation System Layers

As shown in Table 1, the embedding operation during inference has a very low OI compared to other operations such as MLP and feature interaction. Since embedding in inference/training constitutes simple element-wise addition along the dimension, the operations of the embedding layer are memory-bound in CPU or GPU. This means that system performance can be significantly improved by offloading the embedding to CMS, with its high memory capacity and bandwidth. Such implication is in line with the expectations that drove us to choose recommendation as the target application to evaluate our solution.

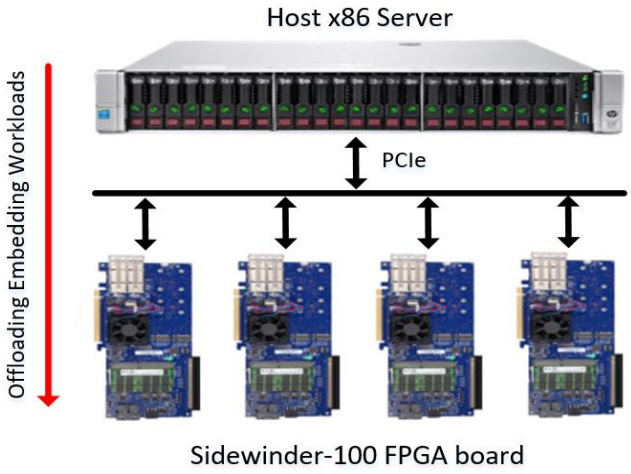


Figure 7. System Configuration of FPGA PoC

The system configuration used for evaluation is shown in Figure 7. A total of four Zynq UltraScale+ Sidewinder-100 FPGA cards that mimic computational memory are connected to an Intel server via PCIe Gen3. The host CPU offloads the embedding layer of the recommendation system to some or all of the four FPGA cards.

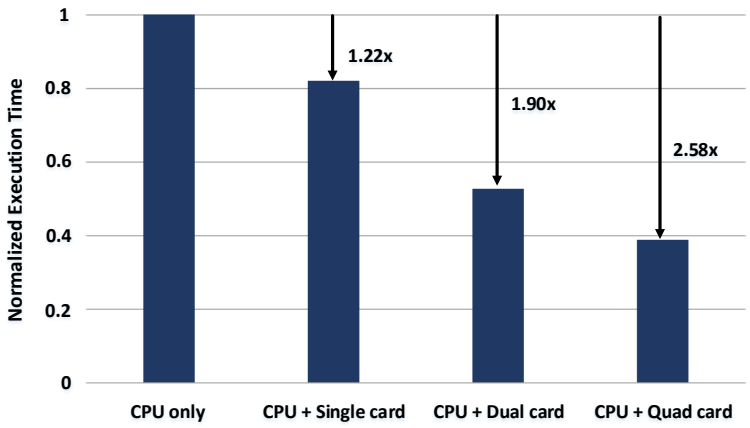


Figure 8. Overall System Performance of FPGA PoC

We set the synthetic dataset condition with three feature tables, 100 pooling vectors per feature, 256 dimensions, and 512 batch size for evaluation. Figure 8 depicts the subsequent performance measurements for the recommendation system using multiple CMS FPGAs. The CPU and single-card combination shows performance gains over a CPU-only system via embedding layer acceleration. Overall performance improves with the number of cards, reaching a 2.58x enhancement on a quad-card configuration compared to a CPU-only system.

By means of target application analysis and system evaluation, we have confirmed the functionality and feasibility of our solution. We note that since the embedding pooling size of the synthetic dataset is 100, data size is reduced by about 100x per card. With a larger pooling size in an actual dataset, system power consumption in data movement is expected to improve further.

Conclusion & Future Work

We have presented an NDP-enabled, novel CMS in this whitepaper. The proposed design provides scalable memory bandwidth and capacity for extreme-scale data-centric workloads. Our design also addresses the data movement bottleneck by adopting an optimized NDP core and expanding the memory node capacity. The experimental results show that, compared to a legacy CPU system, our design presents up to 2.58x improvement in execution time.

While we have targeted data-intensive workloads in recommendation systems for this initial research, we plan to expand the scope of application to data analytics applications. According to our initial experiment that analyzed the number of computations and memory accesses in data analytics operations, such operations require very little computation compared to a large amount of memory access. This implies that data analytics operations are also data-intensive workloads, for which our proposed solution should be highly effective as well. Based on this implication, we will propose a method for processing representative data analytics operations using CMS in the near future.

Reference

[1] S. Polfliet, F. Ryckbosch, L. Eeckhout, “Optimizing the Datacenter for Data-Centric Workloads” In 2011 Proceedings of the international conference on Supercomputing, pp. 182-191, May 2011.

[2] B. Dally, “Power, programmability, and granularity: The challenges of exascale computing,” in Parallel Distributed Processing Symposium (IPDPS), 2011 IEEE International, pp. 878–878, May 2011.

[3] Fedus, William, Barret Zoph, and Noam Shazeer. "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity." arXiv preprint arXiv:2101.03961 (2021).

[4] https://github.com/openai/gpt-3

[5] S. Wang and E. Ipek, “Reducing Data movement Energy via Online Data Clustering and Encoding” In Proceedings of the International Symposium on Microarchitecture (MICRO), Oct 2016.

[6] A. Boroumand, S. Ghose, Y. Kim, R. Ausavarungnirun,E. Shiu, R. Thakur, D. Kim,A. Kuusela, A. Knies, P. Ranganathan, and O. Mutlu, “Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks,” in ASPLOS, 2018.

[7] “Top ten exascale research challenges, DOE advanced scientific computing advisory committee (ASCAC) subcommittee report,” tech. rep., U.S. Department of Energy, February 2014.

[8] V. Seshadri and O. Mutlu, “The processing using memory paradigm: In-dram bulk copy, initialization, bitwise and and or,” arXiv preprint arXiv:1610.09603, 2016.

[9] R. Nair, S. F. Antao, C. Bertolli, P. Bose, J. R. Brunheroto, T. Chen, C.-Y. Cher, C. H. Costa, J. Doi, C. Evangelinos, et al., “Active memory cube: A processing-in-memory architecture for exascale systems,” IBM Journal of Research and Development, vol. 59, no. 2/3, pp. 17–1, 2015.

[10] V. Seshadri, K. Hsieh, A. Boroum, D. Lee, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, “Fast bulk bitwise and and or in dram,” IEEE Computer Architecture Letters, vol. 14, no. 2, pp. 127–131, 2015.

[11] Hoffer, Barak, et al. "Experimental demonstration of memristor-aided logic (MAGIC) using valence change memory (VCM)." IEEE Transactions on Electron Devices 67.8 (2020): 3115-3122

[12] Joonseop Sim, et al. "LUPIS: Latch-up based ultra efficient processing in-memory system." 2018 19th International Symposium on Quality Electronic Design (ISQED). IEEE, 2018.

[13] Joonseop Sim, et al. "Mapim: Mat parallelism for high performance processing in non-volatile memory architecture." 20th International Symposium on Quality Electronic Design (ISQED). IEEE, 2019.

[14] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, “Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories,” in Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE, pp. 1–6, IEEE, 2016.

[15] M. Gao, G. Ayers, and C. Kozyrakis. Practical near-data processing for in-memory analytics frameworks. In 2015 International Conference on Parallel Architecture and Compilation (PACT), pages 113–124, 2015.

[16] Samuel Williams, Andrew Waterman, and David Patterson. Roofline: An insightful visual performance model for multicore architectures. Commun. ACM, 52(4):65–76, April 2009.

[17] M. Smelyanskiy. Zion: Facebook next- generation large memory training platform. In 2019 IEEE Hot Chips 31 Symposium (HCS), pages 1–22, 2019.

[18] Bilge Acun, Matthew Murphy, Xiaodong Wang, Jade Nie, Carole-Jean Wu, and Kim Hazelwood. Understanding training efficiency of deep learning recommendation models at scale. In 2021 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2021.

[19] Evangelos Georganas, Sasikanth Avancha, Kunal Banerjee, Dhiraj Kalamkar, Greg Henry, Hans Pabst, and Alexander Heinecke. Anatomy of high-performance deep learning convolutions on simd architectures. In SC18: International Conference for High Performance Computing, Networking, Storage and Analysis, pages 830–841. IEEE, 2018.

[20] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. ACM Comput. Surv., 52(1), February 2019.

[21] 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.

[22] Liu Ke, Udit Gupta, Carole-Jean Wu, Benjamin Youngjae Cho, Mark Hempstead, Brandon Reagen, Xuan Zhang, David Brooks, Vikas Chandra, Utku Diril, et al. Recnmp: Accelerating personalized recommendation with near-memory processing. arXiv preprint arXiv:1912.12953, 2019.

[23] Udit Gupta, Samuel Hsia, Vikram Saraph, Xiaodong Wang, Brandon Reagen, Gu-Yeon Wei, Hsien-Hsin S Lee, David Brooks, and Carole-Jean Wu. Deeprecsys: A system for optimizing end-to-end at-scale neural recommendation inference. arXiv preprint arXiv:2001.02772, 2020.

About SK hynix Inc.

SK hynix seeks to propel the semiconductor industry forward with global tech leadership, and provide a future of greater value to stakeholders to create a better world with information and communication technology. As the world's third largest chipmaker with know-how and customer trust built over more than 38 years, SK hynix continues delivering on a comprehensive range of memory semiconductor solutions from DRAM and NAND Flash to CMOS image sensors.

The company's advanced memory technologies are driving critical innovations of the Fourth Industrial Revolution such as Big data, AI, Machine Learning, IoT, and Robotics. Moreover, SK hynix is aiming higher with the new "Mem4EST (Memory Forest)" initiative to quickly respond to future changes in the ICT ecosystem. With robust ESG management that accounts for value to the environment, societies, and future generations, SK hynix will continue to build competence and success around the globe.

 W-CMS-E01-210910-R01