Integrating New Photonic-Based Heterogeneous Memory into Throughput Accelerators

Jie Zhang[1] and Myoungsoo Jung[2]
Computer Architecture and Memory Systems Laboratory,
Peking University[1], Korea Advanced Institute of Science and Technology (KAIST)[2]
http://camelab.org

I. Introduction

Graphics processing units (GPUs) have been widely adopted as an efficient accelerator hardware platform to speed up the execution of large-scale data-intensive applications. While massively parallel computing power of a GPU can enhance data processing bandwidth, its memory system is difficult to satisfy increasing I/O demands of the large-scale applications. Specifically, DRAM faces many practical challenges to scale their technology down, and it cannot be denser due to memory retention time violations, insufficient sensing margins and low reliability issues [1].

To address these challenges, we propose *Ohm-GPU*, a new optical network based heterogeneous memory design for GPUs. Specifically, Ohm-GPU can expand the memory capacity by combing a set of high-density 3D XPoint and DRAM modules as heterogeneous memory. To prevent memory channels from throttling throughput of GPU memory system, Ohm-GPU replaces the electrical lanes in the traditional memory channel with a high-performance optical network. However, the hybrid memory can introduce frequent data migrations between DRAM and 3D XPoint, which can unfortunately occupy the memory channel and increase the optical network traffic. To prevent the intensive data migrations from blocking normal memory services, Ohm-GPU revises the existing memory controller and designs a new optical network infrastructure, which enables the memory channel to serve the data migrations and memory requests in parallel. Our evaluation results reveal that Ohm-GPU can improve the performance by 27%, compared to the baseline optical network based heterogeneous memory system.

II. Related Work and Challenges

Baseline GPU Architecture. Figure 1 shows a baseline GPU architecture, which is similar to the real-world GPU products. Specifically, the baseline GPU consists of multiple streaming multiprocessors (SMs), shared L2 cache and memory controllers, all of which are connected through an interconnect network. Within the SMs, a group of 32 threads, called *warp*, are executed in a lockstep. During the execution, a set of instructions for each warp is fetched from the underlying GPU memory. The instructions are then decoded and stored in the register files. Afterwards, the warp scheduler schedules the warps to execute. Arithmetic instructions are executed by ALUs, while load/store instructions generate memory requests. SMs firstly try to find out data associated with the memory requests from L1D cache. If L1D cache misses, the requests will be forwarded to the shared L2 cache via the interconnect network. If L2 cache also misses, the requests will be sent to the memory controller. A traditional GPU memory controller in practice buffers and schedules incoming memory requests. The memory controller issues the memory transactions with DRAM via GDDR6 protocol.

Challenges in GPU Memory System. The existing GPU memory system becomes the performance bottleneck of executing large-scale applications in the GPU due to its low capacity, limited throughput, and high energy consumption [4]–[6]. As the GPU on-chip DRAM cannot accommodate all data sets of large-scale applications, the data often require being loaded/stored from/to an external storage. To be precise, we evaluated a computing system that integrates a high-performance GPU and an SSD together. Figure 2a shows a breakdown analysis to execute different GPU applications on our testbed system. The storage access delay and data transfers between

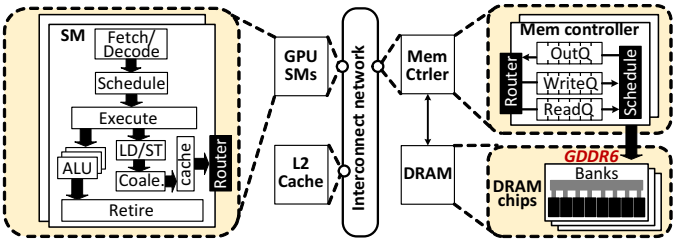
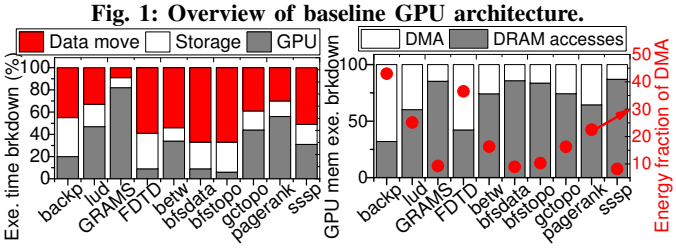


Fig. 1: Overview of baseline GPU architecture.



(a) GPU-SSD integrated system. (b) GPU memory subsystem.

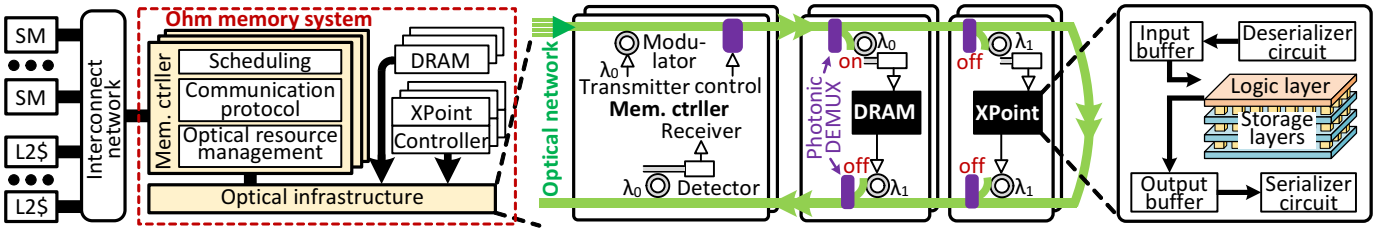
Fig. 2: Breakdown analysis of executing GPU apps.

the GPU and SSD account for 21% and 45% of the total execution time, on average, respectively. This data movement overhead takes a time longer than the GPU computing time itself by 2.3×, on average. We also analyze the impact of DMA and DRAM accesses on the GPU memory system in terms of execution time and energy consumption, and the results are shown in Figure 2b. Transferring data via electrical memory channels (i.e., DMA) degrades the performance of the GPU memory system by 31% and 19% in terms of execution time and energy consumption, respectively.

III. Ohm-GPU Architectural Design

Figure 3a shows an overview of our baseline Ohm-GPU design. Compared to the traditional GPU (cf. Figure 1), Ohm-GPU integrates a new memory system, called *Ohm memory system*, to replace the existing DRAM-based GPU memory system. Specifically, the Ohm memory system employs DRAM and XPoint as a heterogeneous memory to increase the memory capacity while maintaining high performance. DRAM in Ohm-GPU also accommodates write-intensive data, which can significantly reduce the number of writes on XPoint, thereby extending the lifetime of XPoint. To improve the bandwidth and energy consumption behaviors of the memory system, Ohm-GPU also integrates an optical infrastructure, which jointly connects the memory controllers and the memory devices.

Optical infrastructure for Ohm-GPU. Figure 3b shows a high-level overview of our optical infrastructure design. An optical channel replaces hundreds of electrical lanes to connect between the GPU memory controllers and memory devices. Directly attaching multiple memory controllers to a single optical channel can introduce channel-level conflicts, as these memory controllers can compete to occupy the same optical channel. To address this challenge, we statically split the optical channel into multiple virtual channels and assign a dedicated virtual channel to each memory controller. While the virtual channels ensure no channel conflicts among all memory controllers, the transmitters and receivers of massive memory devices may compete to occupy a virtual channel. To address this, we leverage



(a) Ohm-GPU architecture. (b) Optical network based memory channel. (c) New XPoint design.

Fig. 3: Overview of Ohm-GPU architecture with an Ohm memory system.

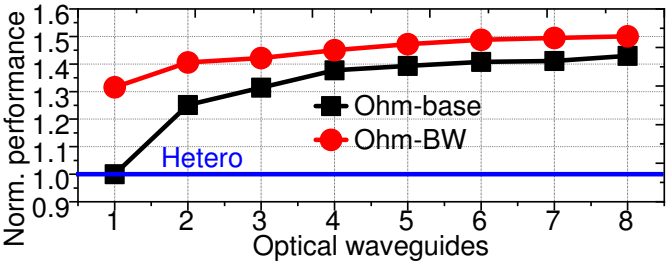


Fig. 4: Performance of the evaluated GPU platforms.

the control logic and photonic demultiplexers proposed in [2] to arbitrate the competition of an optical channel usage.

Integration of DRAM and XPoint. Unfortunately, DRAM and XPoint cannot be directly attached to the optical channel via the photonic transmitters and receivers. There are two reasons. Firstly, the command, address, and data are simultaneously accessed in the memory devices while all the data are serialized in the optical channel. Secondly, XPoint requires assistance of a XPoint controller to enable ECC, manage the endurance of XPoint, and control its I/O transactions. To make the XPoint and DRAM compatible with the optical channel, we employ a SerDes circuit to transform data between serial and parallel I/Os (cf. Figure 3c). We also employ a small piece of registers (i.e., 16KB) in front of the memory devices to buffer the data from the optical channel. To integrate XPoint in the optical channel, one simple solution is to employ a XPoint controller for each XPoint. However, having multiple XPoint controllers can increase the area cost, which is critical as the limited GPU space is concerned. Since XPoint stacks its storage cores into multiple layers, we can save the area cost by integrating the XPoint controller in XPoint as a logic layer, which is adopted by several prior logic-in-memory designs [3].

System Design for Migration Overhead Removal. We observe that data migration between DRAM and XPoint increases the average memory access latency by 51%, owing to two main reasons. First, the data migrations are expensive as the memory controller should copy all the data to its internal buffer and redirect the data to the target memory module. Second, as our optical channel is shared by both data migration and memory requests, data migration consumes the channel resources, which should be used to serve the memory requests. To reduce such overheads, Ohm-GPU enables XPoint controllers to directly migrate data between DRAM and XPoint. To prevent the memory and XPoint controllers from competing to access the same DRAM, Ohm-GPU implements a conflict detection mechanism in the memory controllers, which can detect the potential conflicts before scheduling the memory requests and data migration requests. To achieve full utilization of the optical channel, we create dual routes in the same optical channel to simultaneously serve the memory requests and the data migration tasks. Our new design requires minor optical hardware costs and does not increase the total energy consumption of the target memory system.

IV. Evaluation and Conclusion

Experiments. We implement Ohm-GPU atop a GPU simulator (MacSim). To explore a full design space of optical network based heterogeneous memory subsystems, we replace six 32-bit electrical memory channels with a single optical channel as default. This optical channel configuration can provide the same bandwidth as the traditional electrical memory channels. We implement three different GPU platforms: (1) `Hetero`: a baseline GPU architecture employing an *electrical channel* integrated heterogeneous memory system; `Hetero` leverages the memory controller to migrate data between DRAM and XPoint; (2) `Ohm-base`: a baseline GPU architecture employing an *optical network* integrated heterogeneous memory system; (3) `Ohm-BW`: compared to `Ohm-base`, it integrates the system design for migration overhead removal.

Performance. While a single optical waveguide can achieve the bandwidth same as electrical memory channels of 192 lanes, Ohm-GPU can employ multiple optical waveguides under the same area constrains as the electrical memory channels. Figure 4 shows the performance improvement brought by multiple optical waveguides in Ohm-GPU. `Ohm-base` with 8 optical waveguides improves the system performance than `Hetero`, by 41%, on average. This is because employing multiple optical waveguides can significantly reduce the DMA latency of the heterogeneous memory system. `Ohm-BW` also benefits from the increased number of optical waveguides. The performance improvement can be 17%.

In this work, we propose Ohm-GPU, a new design of optical network for heterogeneous memory integrated GPU, which can mitigate the impact of data migration on optical channel. Specifically, Ohm-GPU decouples the memory controller from the management of the data migration and leverages the dual routes in optical channel to prevent data migration from occupying the memory channel. Our Ohm-GPU can improve the performance by 181% and 27%, compared to a DRAM-based GPU memory system and the baseline optical network based heterogeneous memory system, respectively.

V. Original Publication

J. Zhang and M. Jung. 2021. Ohm-GPU: Integrating New Optical Network and Heterogeneous Memory into GPU Multi-Processors. IEEE/ACM MICRO. *https://github.com/jiezhang-camel/jiezhang-camel.github.io/blob/master/files/paper5-OhmGPU.pdf*

VI. Acknowledgement

This research is mainly supported by NRF 2021R1AC4001773 and IITP 2021-0-00524 & 2022-0-00117. The work is also supported in part by KAIST start-up package (G01190015), Samsung (G01200447) and Samsung HiPER. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies. Myoungsoo Jung is the corresponding author.

References

[1] Y. Kim, "Architectural techniques to enhance dram scaling," Ph.D. dissertation, CMU, 2015.
[2] Z. Li *et al.*, "Exploring high-performance and energy proportional interface for phase change memory systems," in *HPCA*. IEEE, 2013.
[3] M. M. Shulaker *et al.*, "Monolithic 3d integration of logic and memory: Carbon nanotube fets, resistive ram, and silicon fets," in *International Electron Devices Meeting*. IEEE, 2014.
[4] J. Zhang *et al.*, "Nvmmu: A non-volatile memory management unit for heterogeneous gpu-ssd architectures," in *PACT*. IEEE, 2015.
[5] ——, "Flashgpu: Placing new flash next to gpu cores," in *DAC*, 2019.
[6] ——, "Zng: Architecting gpu multi-processors with new flash for scalable data analysis," in *ISCA*. IEEE, 2020.