Cost-effective LLM accelerator using processing in memory technology

Hyungdeok Lee, Guhyun Kim, Dayeon Yun, Ilkon Kim, Yongkee Kwon and Euicheol Lim
SK hynix, Korea.
{hyungdeok.lee, guhyun.kim, dayeon.yun, ilkon.kim, yongkee.kwon, euicheol.lim}@sk.com

2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits) | 979-8-3503-6146-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/VLSITECHNOLOGYANDCIR46783.2024.10631397

Abstract

Large language model (LLM)-based services continue to improve their performance requires the system with both large memory capacity and high memory bandwidth. For the GPT-3 [1][5] 175 billion model to operate at a minimum, it requires 800GB of storage. In addition, from frequent memory access and limited data reuse also affects memory bandwidth. More powerful memory performance requirements, however, comes with significant costs increase. The expenses associated with operating the necessary equipment and services to handle these capacity and bandwidth requirements are considerable.

SK hynix aims to solve this issue by introducing a processing in memory (PIM) device and PIM based accelerator called AiM [2] and AiMX, respectively. By exploiting true bank-level parallelism, AiM and AiMX are expected to enhance the performance of LLM-based services as a core component of disaggregated system and multi-head attention acceleration. Additionally, AiM also has a potential in on-device AI, in direction of both performance and energy consumption with low batch size and reducing off-chip data movement.

Why in-memory computing accelerator system for large language models

LLMs generally have parameters ranging from tens to hundreds of billions, so they require a huge amount of memory capacity. In addition, the performance of LLMs strictly depends on memory bandwidth. Solving this problem, GPU and AI accelerator manufacturers are developing with higher performance memory, which has resulted in significant capital expenditures (CapEx).

Operating expenditures (OpEx) is another critical issue. For instance, the cost of serving 1k input token and 1k output sequence with GPT-4 is currently estimated at $0.09 [3], so the cost of 10 billion queries is around $1 million, and this cost will increase proportionally with model size as well as the number of users and services.

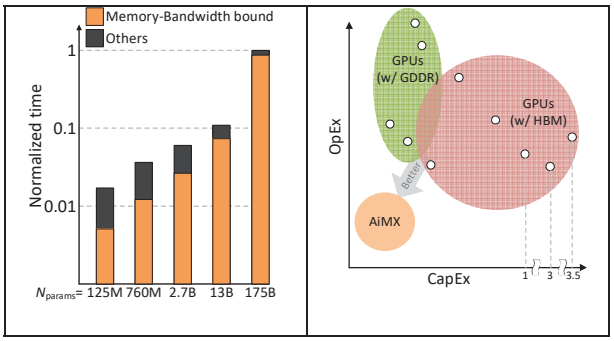


Fig. 1(a) Demonstrate the LLM performance on nVidia V100 GPU. LLM performance is memory bandwidth bound as the LLM size is increase

Fig 1(b) Cost (CapEx and OpEx) comparison of different system architectures. Lower left corner represents cost-

A cost comparison with different approaches to accelerate LLMs is shown in Fig. 1. GPUs with commodity GDDR memory have low CapEx but result in high OpEx. In comparison, GPUs with high bandwidth memory (HBM) reduce OpEx but significantly increase CapEx because of the high manufacturing cost of HBM. Other approaches have similar trade-offs – e.g., AI accelerators can reduce OpEx with specialized, energy-efficient compute engines but introduced significant NRE (non-recurring engineering) cost while still suffering from the memory bandwidth bottleneck. Other in-memory computing approaches have significantly higher CapEx than those based on commodity DRAMs, which have been cost-optimized for decades. To this end, SK hynix introduce in-memory computing accelerator referred to as AiMX, the first LLM accelerator based on AiM which supports true bank parallelism and minimizes data movement – thus, enabling cost reduction of both CapEx and OpEx while maximizing performance. [4]

AiMX architecture and scale-out of AiM system

The system architecture hierarchy of AiMX is shown in Fig. 2 where the processing unit (PU) specialized for LLMs is introduced within each bank of the AiM. A collection of multiple AiMs and an AiM control hub is integrated to create an AiMX (Fig. 3). Note that the AiM control hub includes memory controllers, logics to process the end-to-end operations in LLMs. Multiple AiMX are interconnected together as well as to the CPU host to create an AiMX node and enable a scale-out AiMX system.

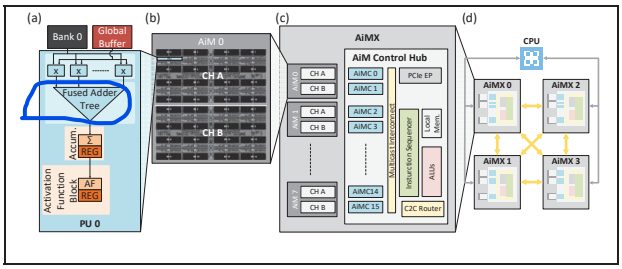


Fig. 2 (a) PU in AiM (b) Single chip of AiM , (c) AiMX, (d) Scale-out AiMX

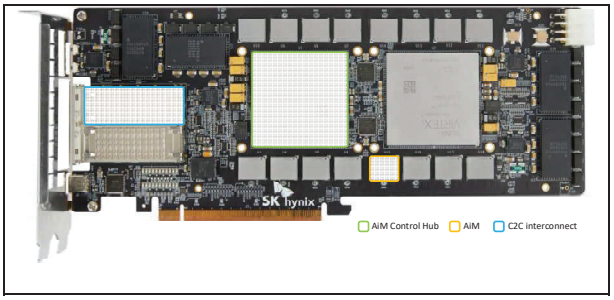
While bank-level parallelism within DRAM is commonly exploited in system architecture, AiM is the first DRAM that exploits "true" all-bank parallelism, enabled by all-bank simultaneous activation to maximize compute efficiency and all-bank parallel computation to maximize parallelism.

The all-bank activation, followed by all-bank parallel compute command, enables a theoretical 16x improvement in in-DRAM bandwidth, leading to high performance and cost-efficiency.

AiM control hub connects multiple AiMs within the card to achieve high performance. In order to fully utilize multiple AiMs, we introduce specially designed scalable instructions.

In particular, a multicast (command) interconnect is designed to enable efficient scaling and CISC-like instructions are introduced to amortize the cost of the command (or instruction) overhead so that single instruction can be broadcasted across all of the channels and across all the banks

within each channel at peak command bandwidth. A custom Card-to-card (C2C) interconnection is supported for further performance expansion. In addition AiM control hub includes



Softmax and layer normalization to make LLM easier to accelerate.

Rationale to adopt a disaggregated system architecture in LLMs

LLM based service, such as chat-bot or question and answer can be divided into two stages – a prefill stage and a response stage. In the prefill stage, which focuses on comprehending questions, most operations in LLMs are expressed as compute-bounded matrix-matrix multiplication requiring high throughput computing system. On the other hand, in the response stage, which involves generating answers to these questions, most of the operations are expressed as memory-bounded matrix-vector multiplication requiring high bandwidth memory system. To solve these requirements, we suggest disaggregated system where GPU and AiMX is responsible to the prefill and response stage, respectively. To be specific, the prefill stage, which has a computation-bound characteristic, is handled by GPUs with high throughput tensor cores to maximize its efficiency. At the same time, the memory-bound nature of the response stage is handled by AiMXs through in-memory processing, aiming to benefit not only in terms of performance but also power consumption.

Opportunities as a multi-batch multi-head attention accelerator

Recently, LLM-based service vendors make the efforts to overcome the bottleneck of memory bandwidth and improve compute utilization of GPUs or AI accelerators. One of this attempt is delivering input in batches, converting FC layers from matrix-vector multiplication to matrix-matrix multiplication. As a result, the performance gain achieved by exploding bandwidth with AiM becomes relatively smaller than before.

Fortunately, albeit to batching, different history data for the response stage is accumulated for each token in multi-head attention (MHA) layer. Therefore, it is difficult to share and reuse history data across tokens. As a result, MHA layers in LLMs mainly consist of matrix-vector multiplication and retains the performance improvement seen with AiM. In addition, multi-batch LLMs results in larger processing time portion of MHA layers than single-batch ones.

To back this up, we estimated processing time of MHA layers, FC layers and the others when generating the 2048th token based on GPT-3 13B model with different batch sizes (1 vs. 16) on A100 GPU. In case of single batch, MHA layers and FC layers occupies 6% and 90.1% of the total execution time, respectively. Otherwise, in case of 16 batches, MHA layers and FC layers occupies 50.6% and 47.4% of the total time, respectively. (Fig. 4a).

We also compared the performance of MHA layers on GPUs and AiMXs. We assumes a benchmark with 16 batches and 4k/4k input/output token on GPT3-13B. We have configured two systems to meet the capacity requirements: one with two 80GB A100 cards and the other with four 32GB AiMX cards. As depicted in Fig. 4b, GPU system takes 135.35 seconds to process the MHA layers, the AiMX system only takes 53.37 seconds, which provides 2.5x higher performance.

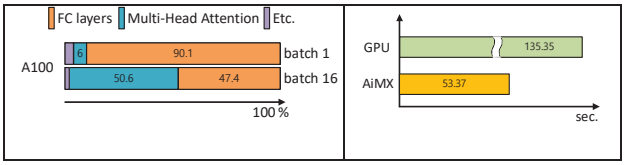


Fig. 4(a) Performance difference between batches by GPU in GPT-3 13B.

AiM opportunities for on device AI services

One of the key distinctions between server/cloud AI services and on-device AI services is batch size. As on-device AI services prioritize individualization and security, they typically use a single-batch. In this regard, AiM can accelerate both FC layers and MHA layers. We estimated the expected performance when AiM replaces DRAM with our in-house analytical model and it shows 6x higher performance or more.

Battery life is another critical concern in on device AI. On device AI services often involve extensive off-chip data movement between DRAM and mobile application processor (AP), leading to substantial energy consumption and significantly reducing battery life. However, AiM effectively addresses this issue by reducing the off-chip data movement through the inclusion of a processing unit within the DRAM itself. By minimizing data transfers, AiM helps to conserve energy and enhance battery performance, providing a solution to this challenge.

Replacing mobile DRAM to AiM is not trivial because AiM must be used as both main memory and accelerator. Some specific concerns to implementing AiM in mobile is as follows:

i) Orchestrating normal memory request from mobile AP and AiM control hub.

ii) Reserving/allocating memory region for AiM operation

iii) Address interleaving for AiM operation

To resolve this issue, we are currently analyzing requirement from mobile system and architecting next version of AiM and AiM control hub targeting mobile system.

References

[1] BROWN, Tom, et al. Language models are few-shot learners. Advances in neural information processing systems, 2020, vol. 33, p. 1877-1901 [2] HE, Mingxuan, et al. Newton: A DRAM-maker's accelerator-in-memory (AiM) architecture for machine learning. In: 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2020. p. 372-385 [3] OpenAI accessed. Feb.05.2024 https://openai.com/pricing [4] KWON, Yongkee, et al. Memory-Centric Computing with SK Hynix's Domain-Specific Memory. In: 2023 IEEE Hot Chips 35 Symposium (HCS). IEEE Computer Society, 2023. p. 1-26 [5] VASWANI, Ashish, et al. Attention is all you need. Advances in neural information processing systems, 2017, vol. 30