KU-The Future

GPU와 PIM 구조의 메모리 주소 매핑 방식에 따른 GEMV 커널 성능 분석

Performance Analysis of GEMV Kernels by GPU and PIM Memory Address Mapping Approaches

신지원ᵒ, 구건재
고려대학교 컴퓨터학과

Outline

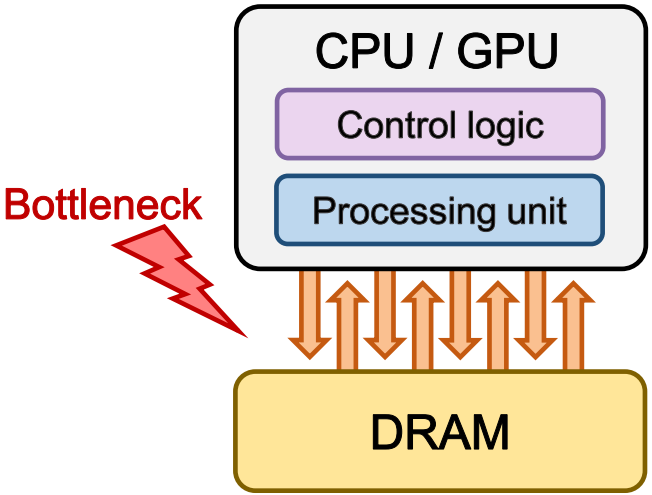
- Introduction/Motivation
- Background
- Experiment
- Conclusion

Outline

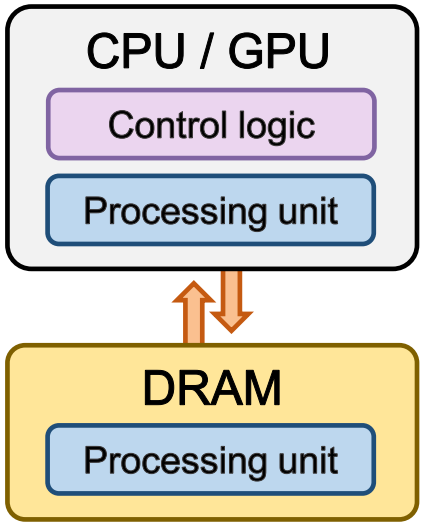
- Introduction/Motivation
- Background
- Experiment
- Conclusion

GPU and PIM (Processing-In-Memory)

- DRAM data movement



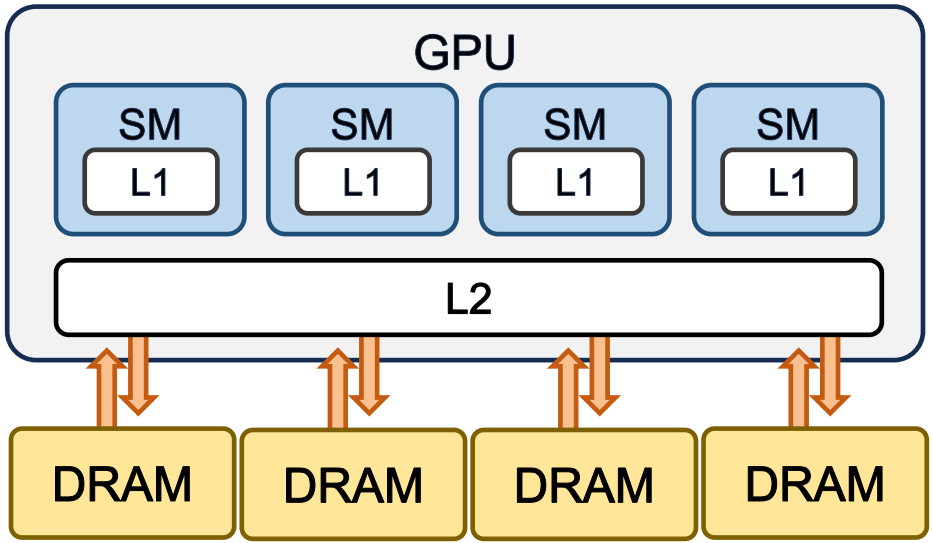
Conventional system



PIM-based system

Data Movement between GPU and PIM

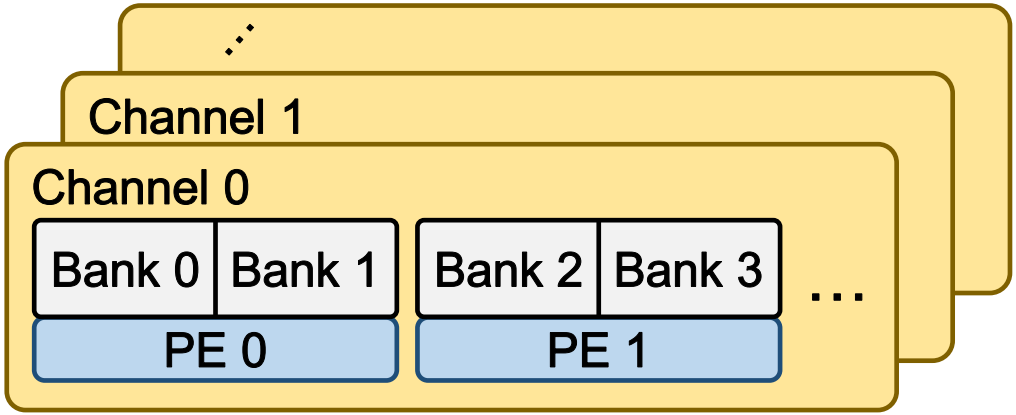
- GPU exploits channel-level parallelism



Overall GPU architecture

Data Movement between GPU and PIM

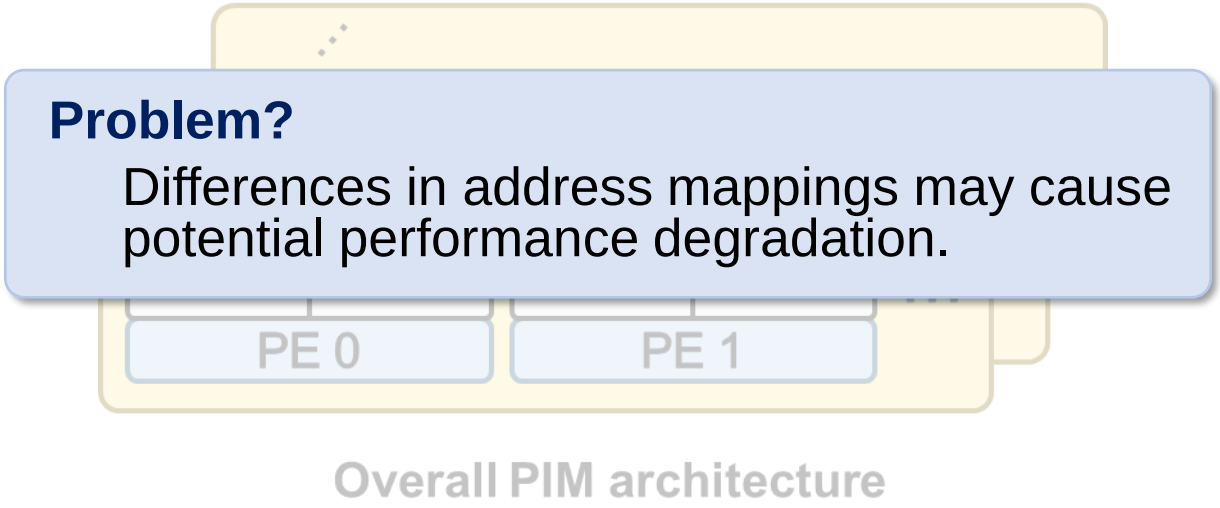
- GPU exploits channel-level parallelism
- PIM exploits bank-level parallelism



Overall PIM architecture

Data Movement between GPU and PIM

- GPU exploits *channel-level* parallelism
- PIM exploits *bank-level* parallelism



Problem?

Differences in address mappings may cause potential performance degradation.

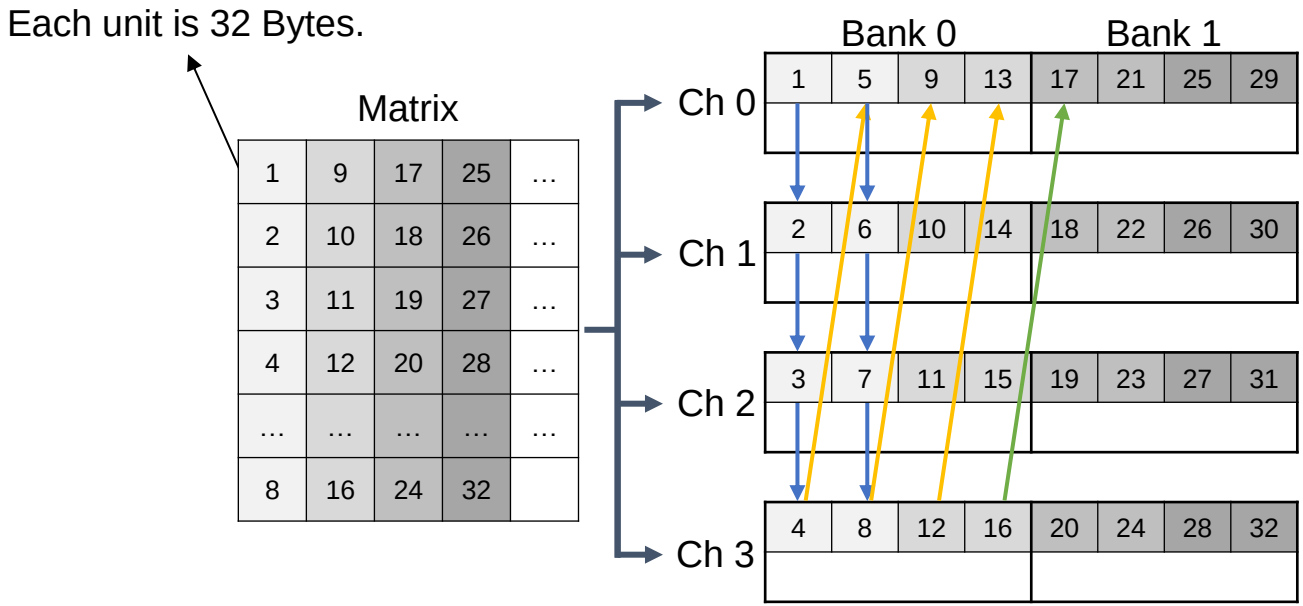
Outline

- Introduction/Motivation
- Background
- Experiment
- Conclusion

Address Mapping – GPU

Ro: row
Ra: rank
Ba: bank
Co: column
Ch: channel

- Address Mapping: Ro(Ra)BaCoCh



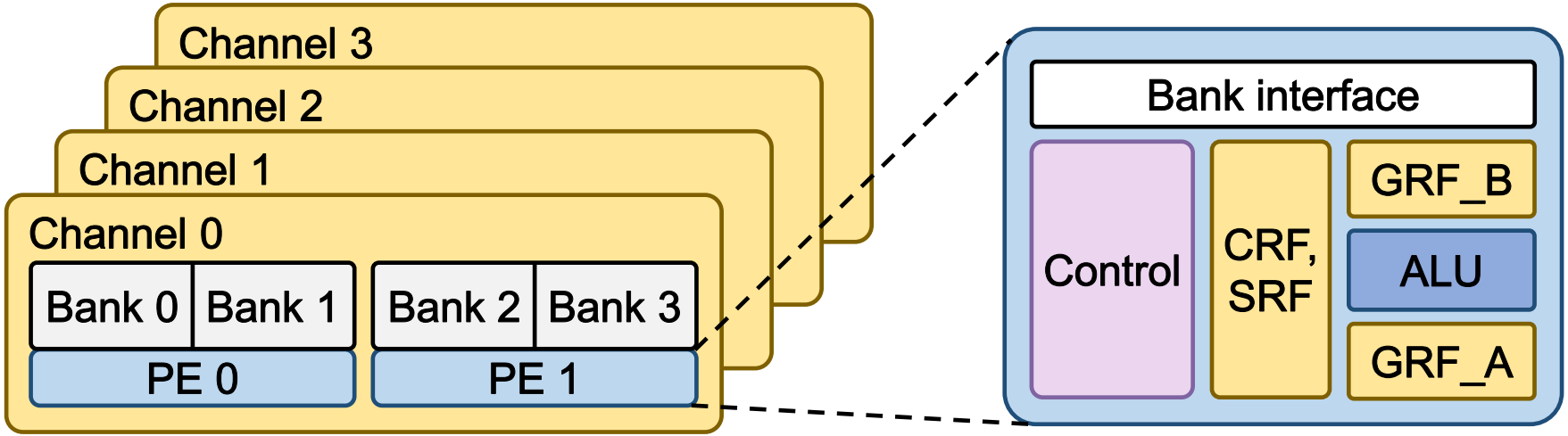
Address Mapping – HBM-PIM

Ro: row
Ra: rank
Ba: bank
Co: column
Ch: channel

- Address mapping: (Ra)RoCoBaCh
- With this mapping, the elements within a matrix are allocated across multiple channels
- Data is reallocated by tiling the matrix in software stack

Address Mapping – HBM-PIM

- The tile size depends on matrix size and hardware configuration

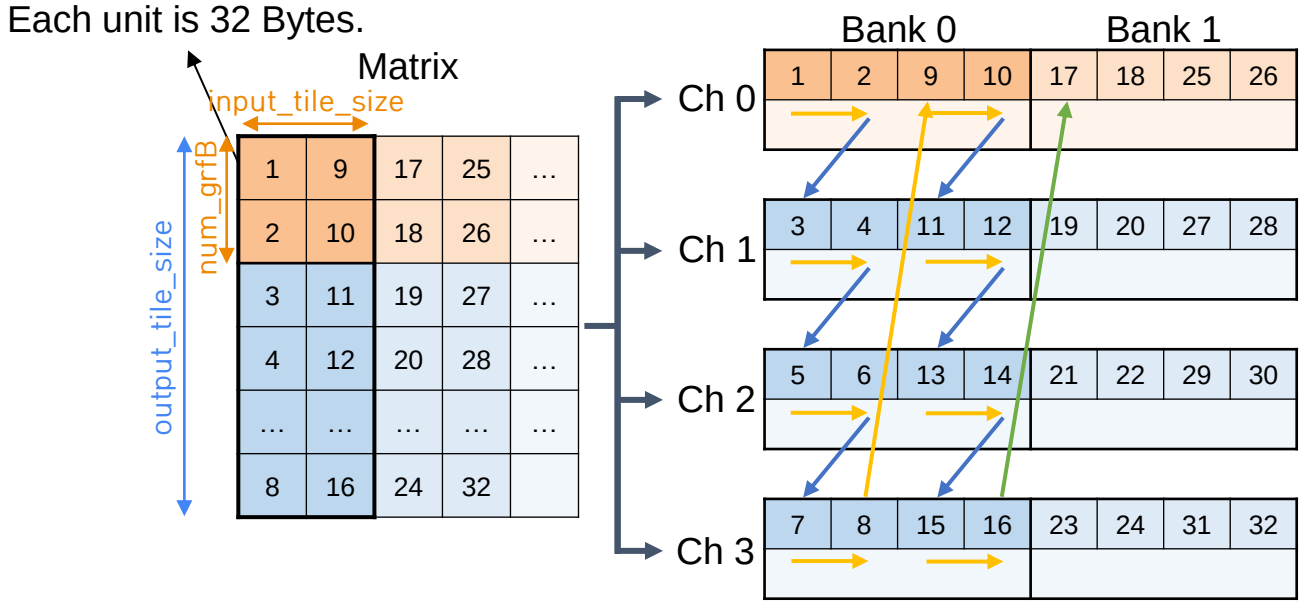


Microarchitecture of HBM-PIM execution unit

Lee, Sukhan, et al. "Hardware Architecture and Software Stack for PIM Based on Commercial DRAM Technology: Industrial Product." 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021.

Address Mapping – HBM-PIM

- Address Mapping: $Ro_{high}Ba_{high}Co_{high}Ro_{low}ChBa_{low}Co_{low}$



Ro: row
Ra: rank
Ba: bank
Co: column
Ch: channel

Address Mapping – HBM-PIM

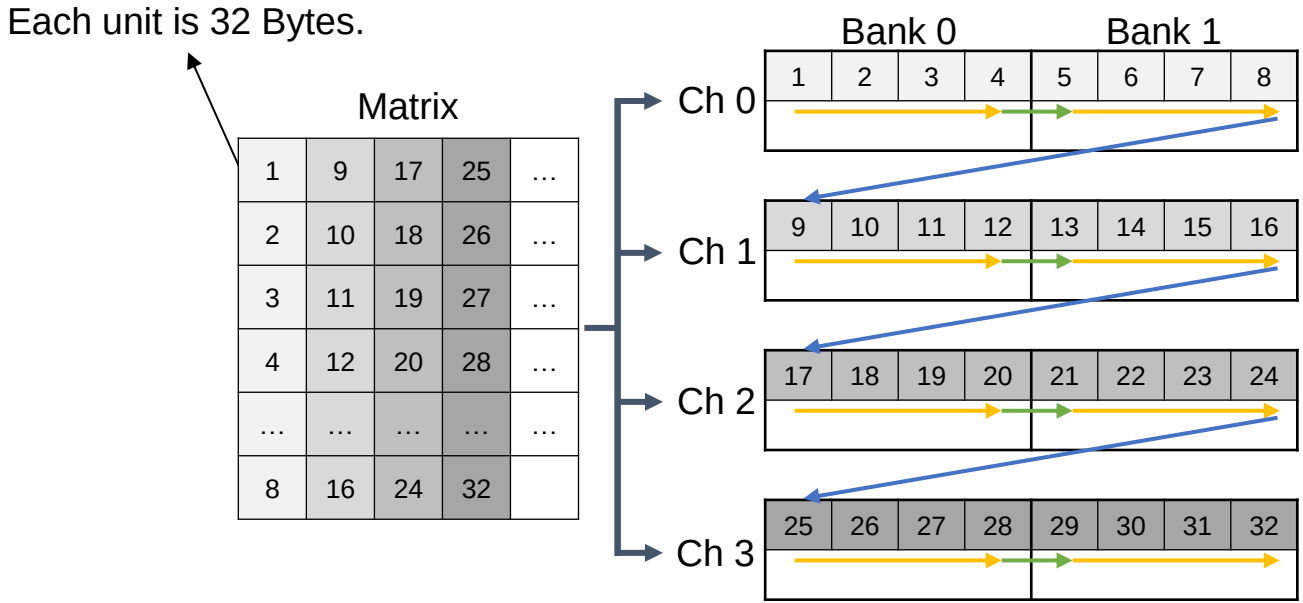
- Address mapping: $Ro_{high}Ba_{high}Co_{high}Ro_{low}ChBa_{low}Co_{low}$
  - Co_{low}: $\log_2 num\_grfB$
  - Ba_{low}: $\log_2 num\_bank - 1$ (1: odd/even bank)
  - Ro_{low}: $\log_2(column\_size\_of\_matrix/out\_tile\_size)$ if column-major

Ro: row
Ra: rank
Ba: bank
Co: column
Ch: channel

Address Mapping – AiM

Ro: row
Ra: rank
Ba: bank
Co: column
Ch: channel

- Address Mapping: RoChBaCo
  - Runtime library may allocate data



Kwon, Yongkee, et al. "Memory-Centric Computing with SK Hynix's Domain-Specific Memory." 2023 IEEE Hot Chips 35 Symposium (HCS). IEEE Computer Society, 2023.

Address Mapping – Summary

Ro: row
Ra: rank
Ba: bank
Co: column
Ch: channel

- GPU: Ro(Ra)BaCoCh
- HBM-PIM: $Ro_{high}Ba_{high}Co_{high}Ro_{low}ChBa_{low}Co_{low}$
- AiM: RoChBaCo

Channel-level parallelism?
GPU > HBM-PIM > AiM

Outline

- Introduction/Motivation
- Background
- Experiment
- Conclusion

Experiment – Methodology

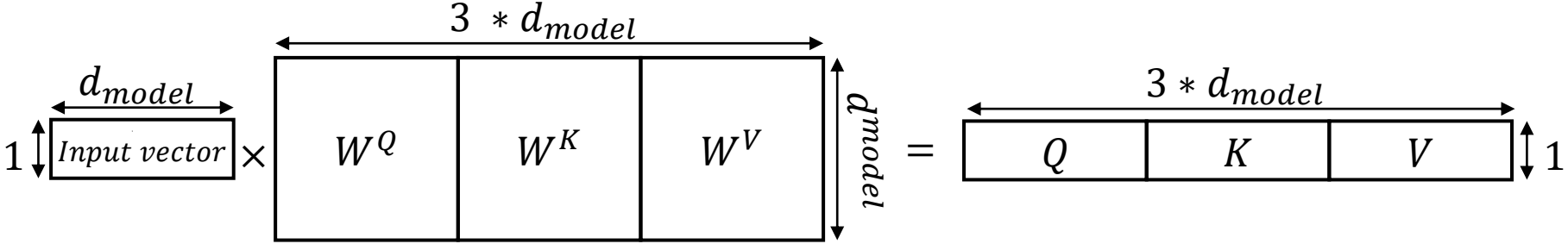
- Simulation with GPGPU-Sim v4.2

Configuration	
GPU	NVIDIA RTX 2060
SM count	34
L1/SM	64 KB
L2 cache/SM	4MB
DRAM	GDDR6, 4 GB
# channels	16
# banks per channel	16
Transaction size	512 B (32 B per channel)

Experiment – Methodology

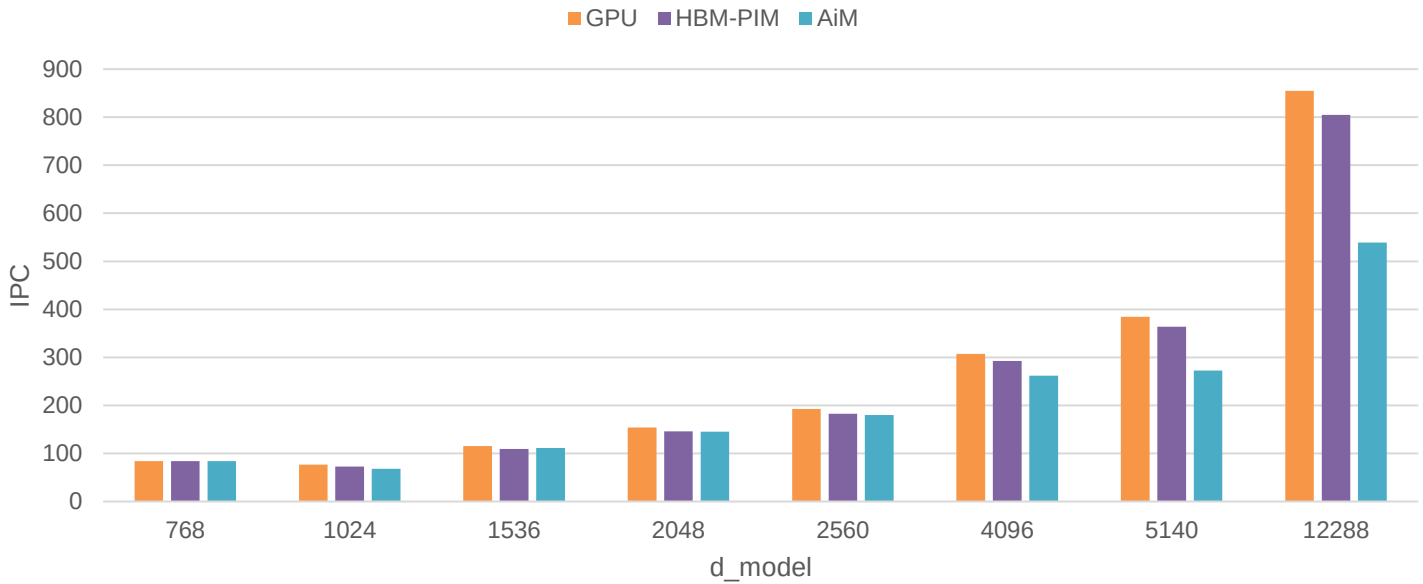
- Transformer GEMV kernel – Q, K, V projection
  - $[Q, K, V] = input\ vector * W$

    $(W = [W^Q, W^K, W^V])$
- In our experiments, d_{model} follows the sizes of GPT-3



Experiment – Result

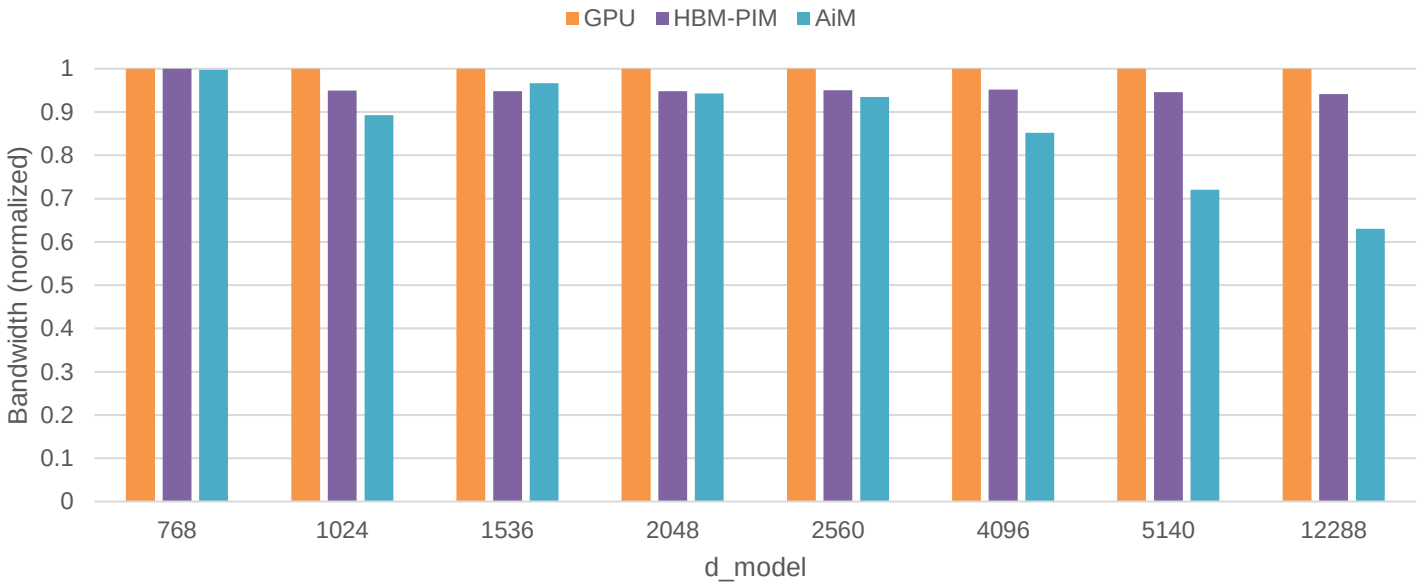
- IPC



HBM-PIM and AiM's address mapping approach results in 4.5% and 13.4% performance drop, respectively, compared to standard GPU addressing.

Experiment – Result

- Bandwidth



HBM-PIM and AiM's address mapping approach use 5.2% and 15.1% less bandwidth, respectively, compared to standard GPU addressing.

Outline

- Introduction/Motivation
- Background
- Experiment
- Conclusion

Conclusion

- Differences between GPU and PIM address mapping schemes can cause performance degradation in GPU-PIM architecture.

- We found performance and bandwidth decreased when using PIM's address mapping approach for GPU using GEMV kernel.

- To optimize performance in GPU-PIM system,
  further research is needed to resolve the performance overhead by the address mapping schemes.

Thank you