Cost effective LLM inference solution using SK hynix's AiM (Accelerator-in-Memory)

Yongkee Kwon, SK hynix

Generative AI based on Large Language Models (LLMs): Challenges and Opportunities

SK hynix's Accelerator-in-Memory (AiM) and AiM-centric accelerator (AiMX) architecture

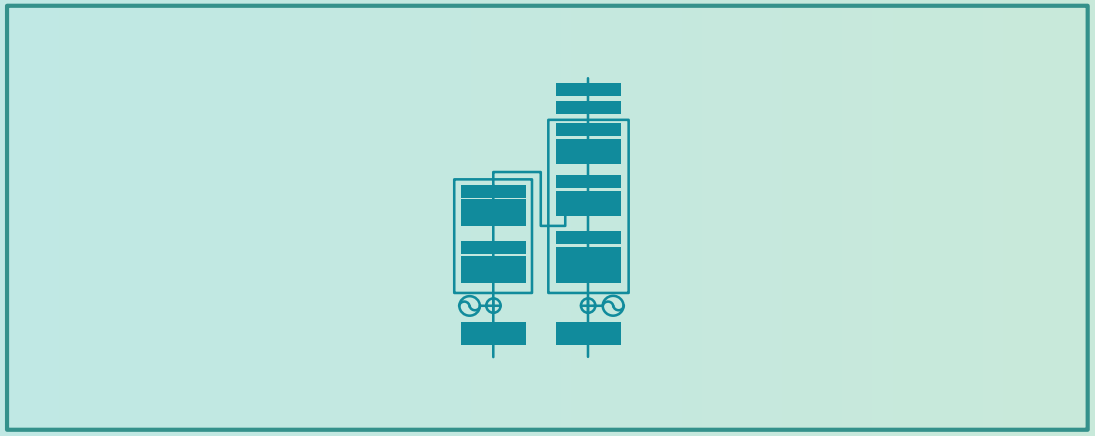
AiMX System-level Performance Analysis and System Integration/Deployment Options

Generative AI and Inference Cost

Generative AI

"This new technology can help people everywhere improve their lives"*

Large Language Models (LLMs)
behind the generative AI boom



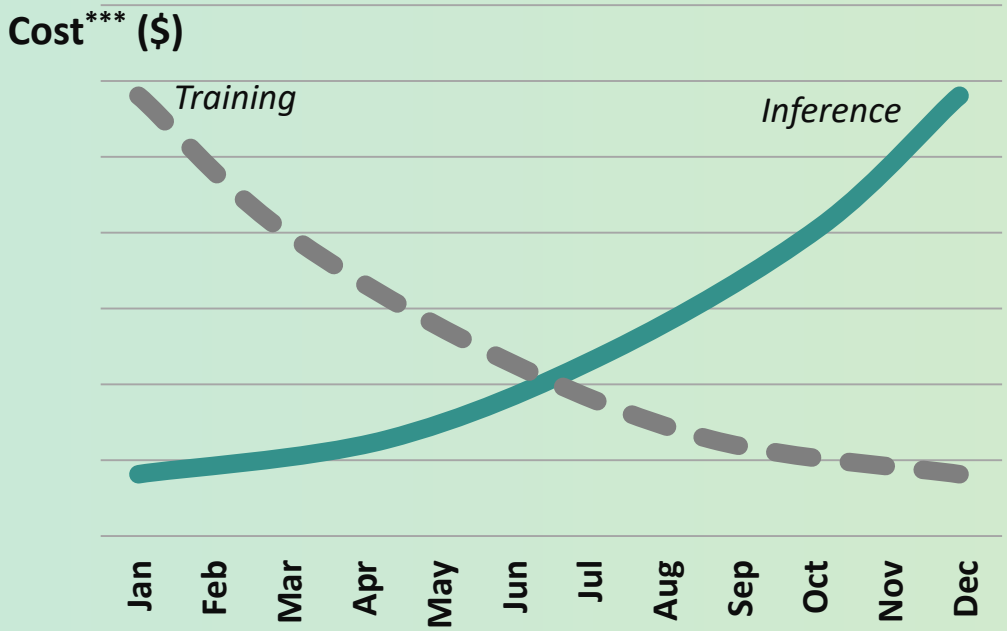
Chat **Code** **Translation** **Search** **Q&A**

(*) *"The Age of AI has begun", Bill Gates, March,2023*

Generative AI Services

"Inference costs eclipses training costs over time"

Inference is all about efficiency
- Performance, Cost**, and Energy -



(**) *TCO (Total Cost of Ownership) ~ CapEx + 3 * OpEX*
(***) *Inference cost ~ #users, assumed to be growing over months*

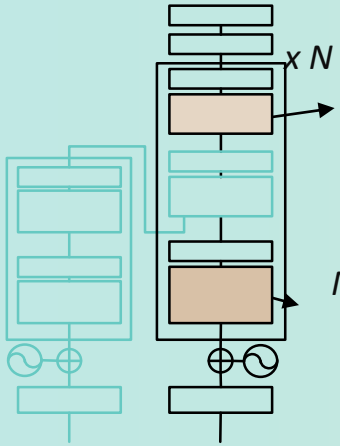
Large Language Model: “It’s the Memory, …”

Transformer Model

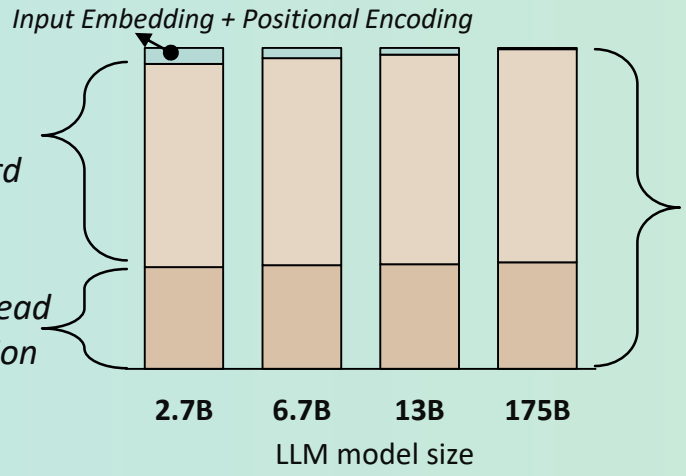
“Fundamental building block of all LLMs”

- Transformer autoregressive decoder*: many large matrix-vector multiplications (or GEMV)

Transformer Architecture



 # parameters (per layer)



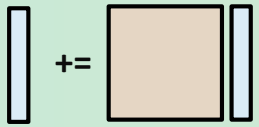
Matrix-Vector Multiplication

“All about moving matrices”

- GEMV: memory BW-bound with low arithmetic intensity
- GEMM: compute-bound with sufficient reuses

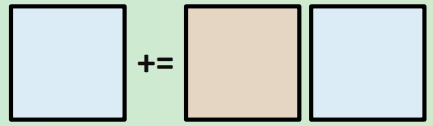
Matrix-Vector Product (GEMV)

$y \leftarrow \alpha Ax + \beta y$



Matrix-Matrix Product (GEMM)

$C \leftarrow \alpha AB + \beta C$



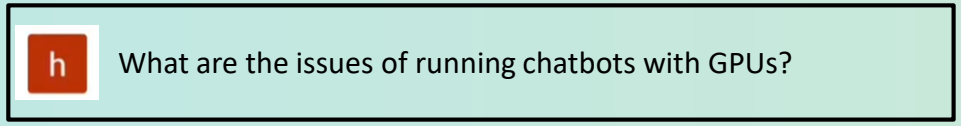
∴ Memory-centric computing for efficient LLM inferences

AI Chatbots: Prompt & Response Characteristics

- AI chatbots consist of input token processing (**prompt**) and answer generating (**response**)
- Especially, **response** stage is heavily memory intensive, as generating one token at a time (autogression)

Prompt: Comprehension

Input: 9 Words / #LLM Model Read(s): 1

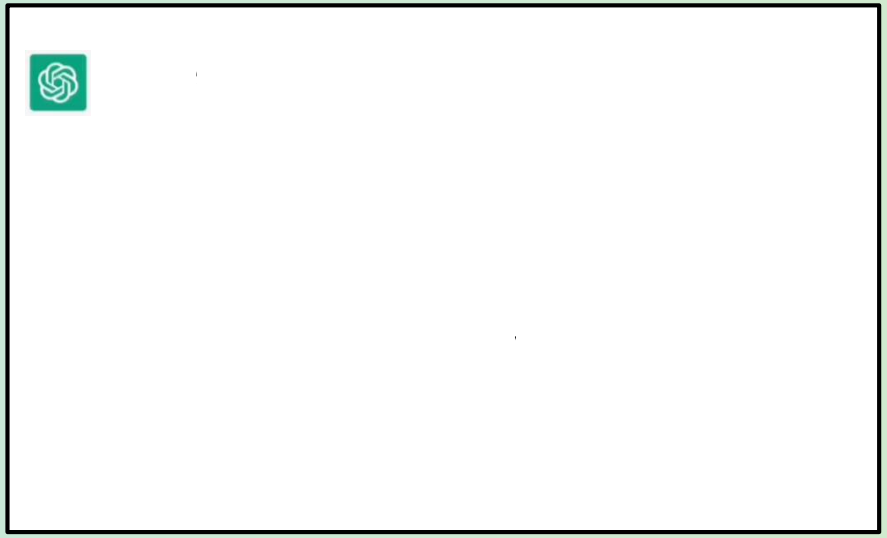


What	are	the
issues	of	running
Chatbot	with	GPUs

Computing-Intensive

Response: Generating Answer

Output: 196 Words / #LLM Model Read(s) : 261



Memory-Intensive

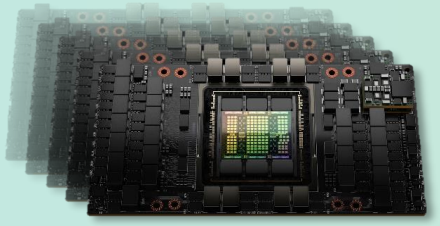
For simplicity's sake, we use 'Word' instead of 'Token'. 1 Token is corresponding to 0.75 Word on average and, therefore, the number of an entire LLM model 'Read's is 261.

State-of-the-art GPU system for AI Chatbot Services

- Need for higher performance and more cost-effective computing infrastrue than current SOTA GPU system

Inference Time with GPU System

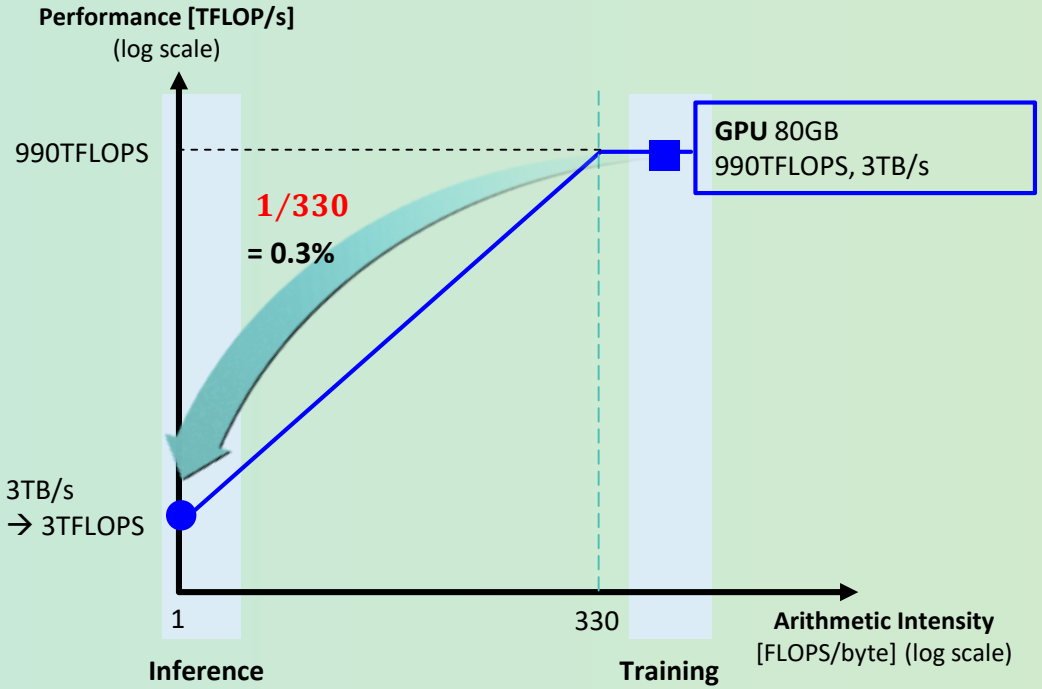
175B generative AI model case



SOTA GPU with HBMs (80GB, 3TB/s) x 5

Model Size	**350 GB**
Bandwidth	**15 TB/s**
Processing Time (1 token)	**23 mSec** (350GB/15TB/s)
Processing Time (261 token)	**6.0 Sec**

Roofline: Training vs. Inference

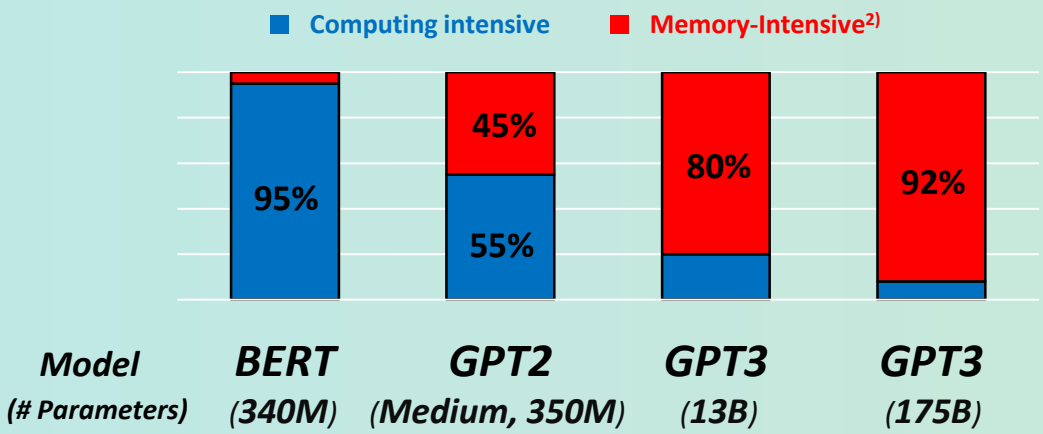


- **Extremely low performance per cost**
  - Exploiting only 0.3% of the peak GPU performance, wasting GPU power consumption and compute capability

Architecting PIM for Cost effective LLM inference solution

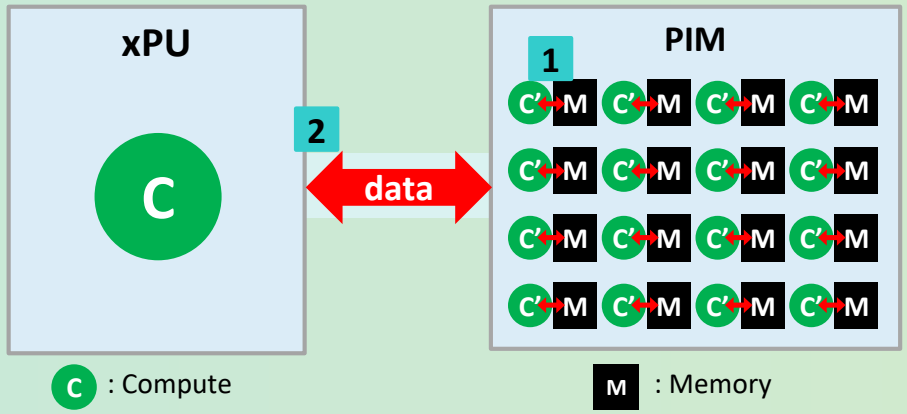
- Processing-in-Memory (PIM) can be architected for highly efficient Generative AI services → SK hynix's AiM

Generative AI service w.r.t. models/sizes



- The larger the model, the **more memory intensive function (specifically, "GEMV"),** so **Memory Bandwidth for GEMV operation** has a greater impact on system performance than the processor

Processing-in-Memory (PIM)



1	**Performance Improvement** By utilizing the higher bandwidth inside the memory
2	**Energy Efficiency Improvement** By minimizing data movement between the host and memory

PIM can be designed to efficiently accelerate memory-Intensive applications such as LLM inferences at relatively low cost

Generative AI based on Large Language Models (LLMs): Challenges and Opportunities

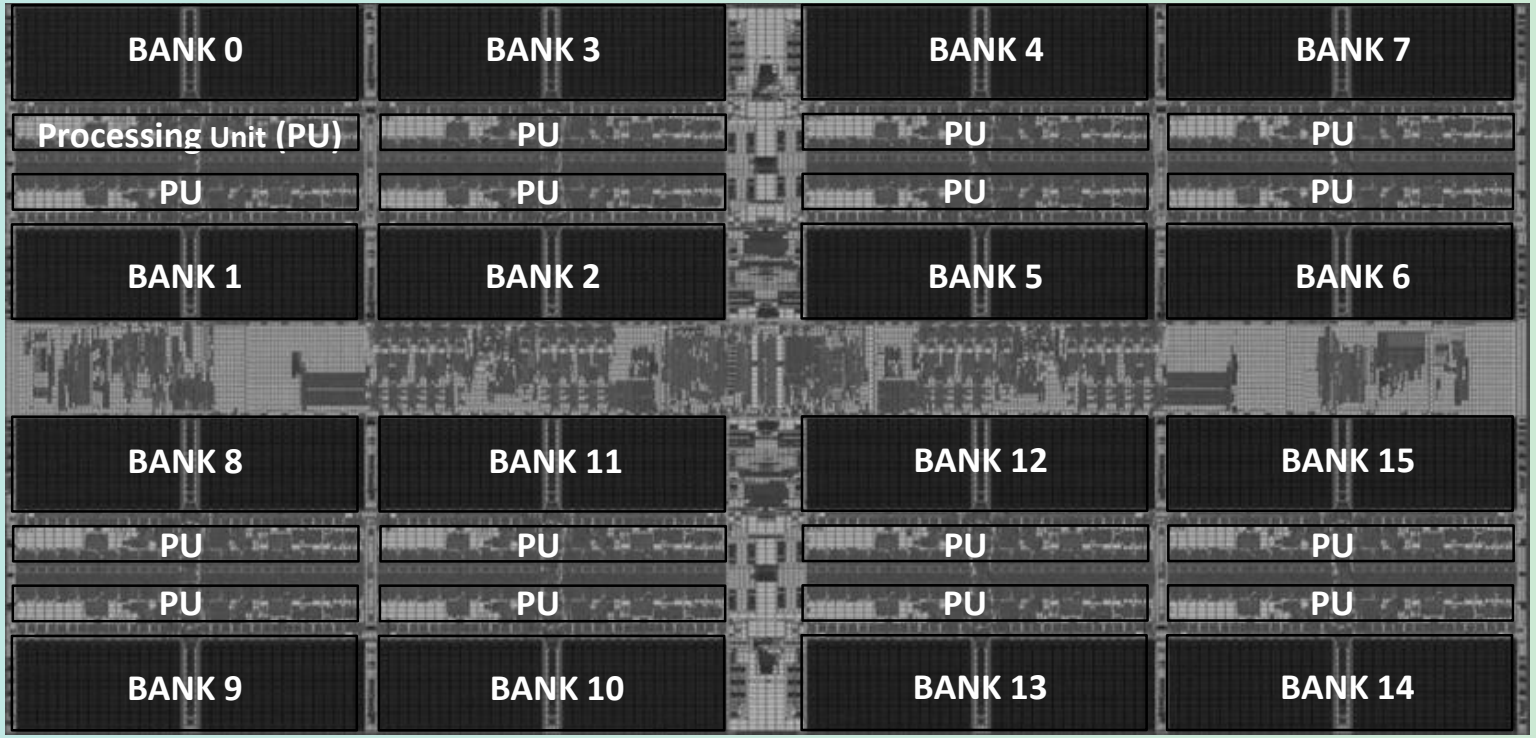
SK hynix's Accelerator-in-Memory (AiM) and AiM-centric accelerator (AiMX) architecture

AiMX System-level Performance Analysis and System Integration/Deployment Options

Accelerator-in-Memory: “True All-Bank Parallelism”

- SK hynix’s First GDDR6-based Processing-in-Memory Product Sample
- Primary Design Goal: No Compromise in Parallelism (Performance= Bandwidth)

GDDR6-AiM Die Photograph



GDDR6-AiM* (per die)	
DRAM Type	GDDR6
Process Technology	1y
Memory Density	4Gb
Organization	X16
IO Data rate	16 Gbs/pin (@1.25V)
(External) Bandwidth**	32 GB/s
Operating Speed	1 GHz
Processing Unit (PU)	16 PU/die
Compute Throughput**	512 GFLOPS
Internal Bandwidth**	512 GB/s
Numeric Precision	BF16
Activation Function support***	Sigmoid, tanh, GELU, ReLU, Leaky ReLU, …

(*) [ISSCC’22] A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications”

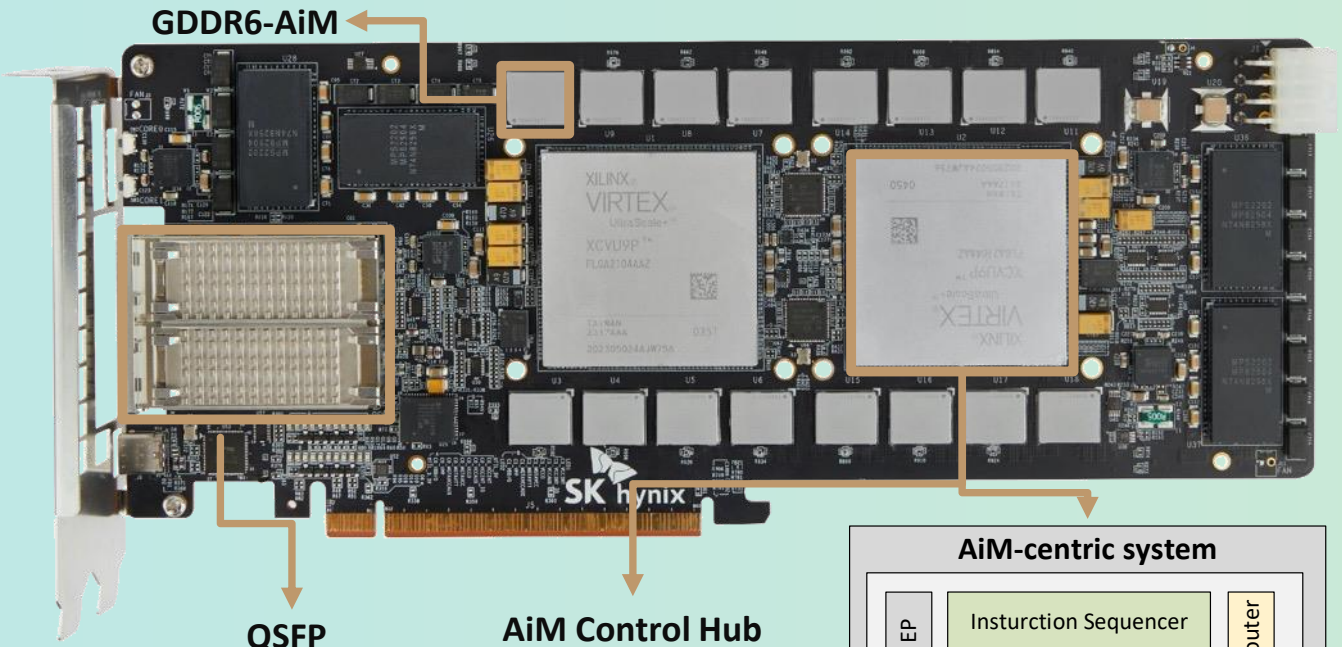
(**) Defined as a peak during burst operations

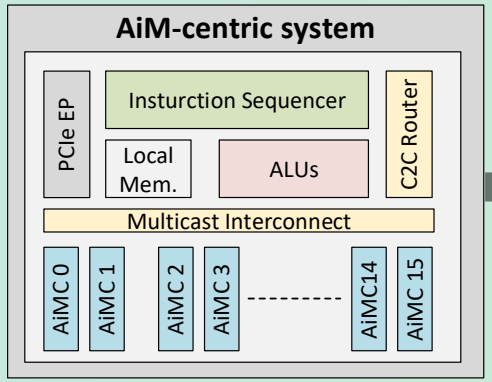
(***) Any customized function may apply with limitation in accuracy using internal lookup table and linear interpolation unit.

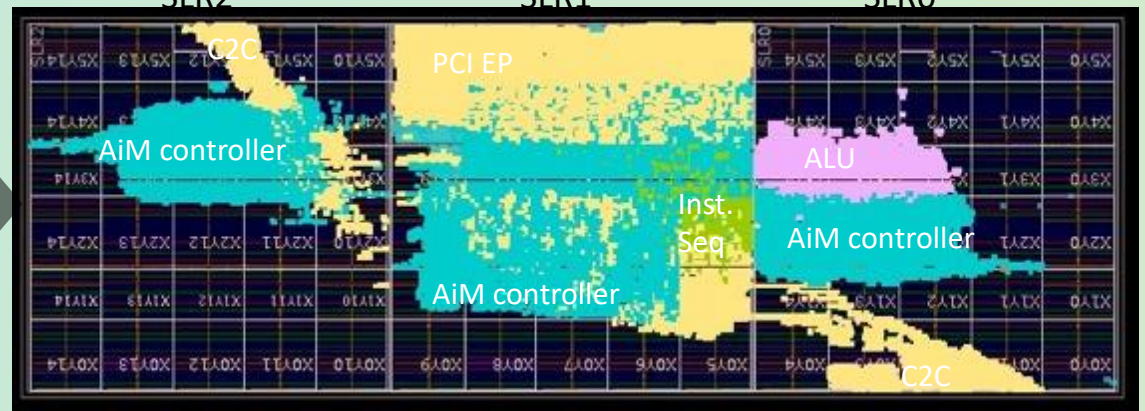
AiMX: Proof-of-Concept of Scale-out AiM System for LLM

- Scale-out AiM realization for proof-of-concept to demonstrate and analyze end-to-end performance, power, and scalability

AiM-centric Accelerator (AiMX) prototype







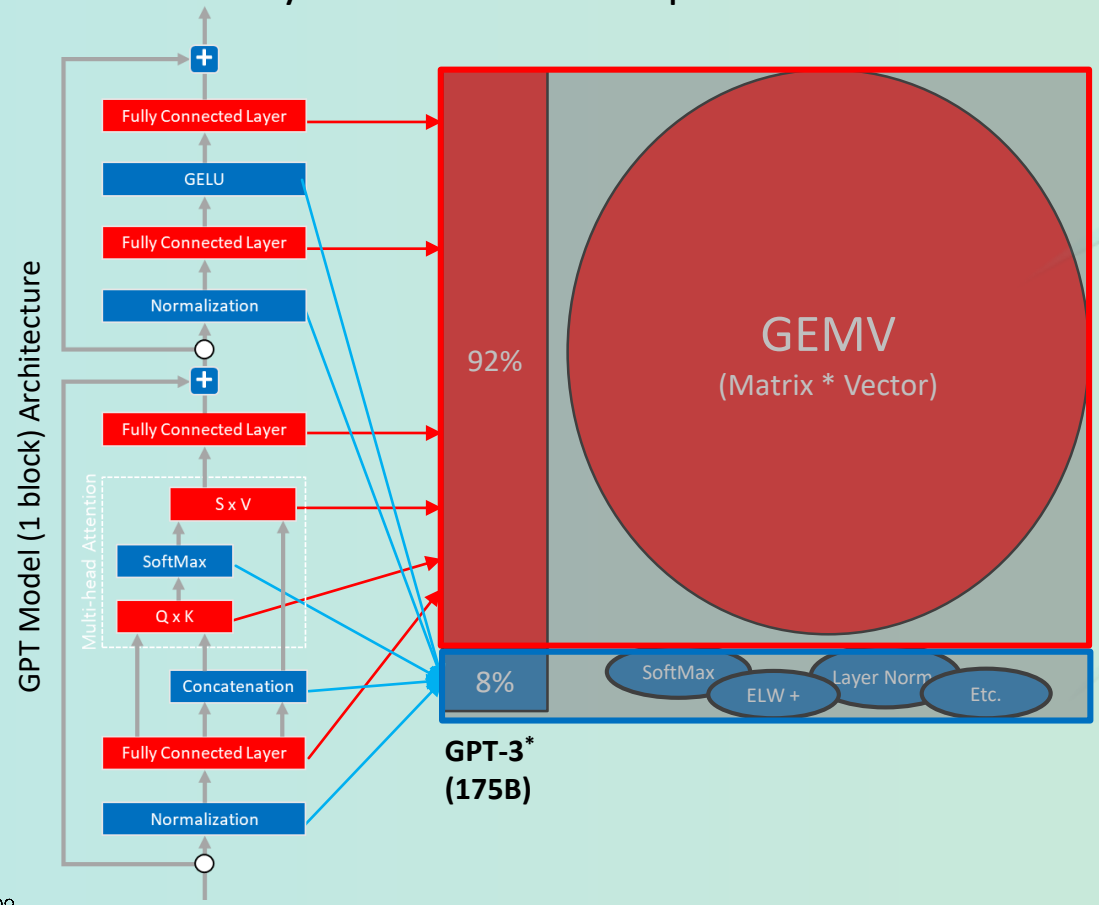
Specification

Host Interface		PCIe Gen3 x8x8 (bifurcated)
Form Factor		FHFL (A100/A30 compatible)
Configuration		2 FPGA* x 16 AiM package
AiM	Capacity	16 GB
	Bandwidth	170 GB/s (@2.67Gbps**)
Scale out		chip2chip interconnect (QSFP28)
Thermal Cooling		Passive

(*) Xilinx Virtex Ultrascale+ (VU9P)
(**) 1/6 of peak data rate of GDDR6, 6Gbps (or 1TB/s)

AiM and AiMX – Efficiency & Flexibility

- Efficiency: AiM chip processes large amount fixed memory-intensive function (GEMV) **efficiently**
- Flexibility: AiM-Control Hub processes small amount various functions **flexibility**

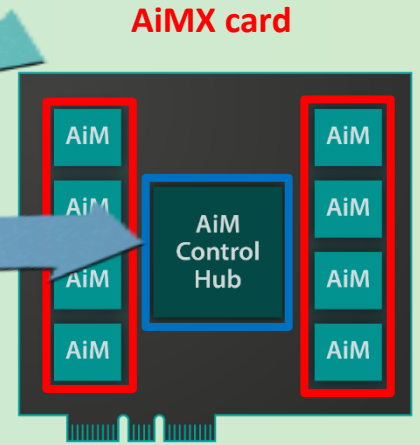


AiM

- Fixed Function (GEMV)
- Memory Intensive
- Large work
- Dedicated unit

AiM Control Hub (ACH)

- Various Functions
- Compute Intensive
- Small work
- Flexible unit



(*) Measured data using 1x NVIDIA V100 GPU with PyTorch (v2.0)

Generative AI based on Large Language Models (LLMs): Challenges and Opportunities

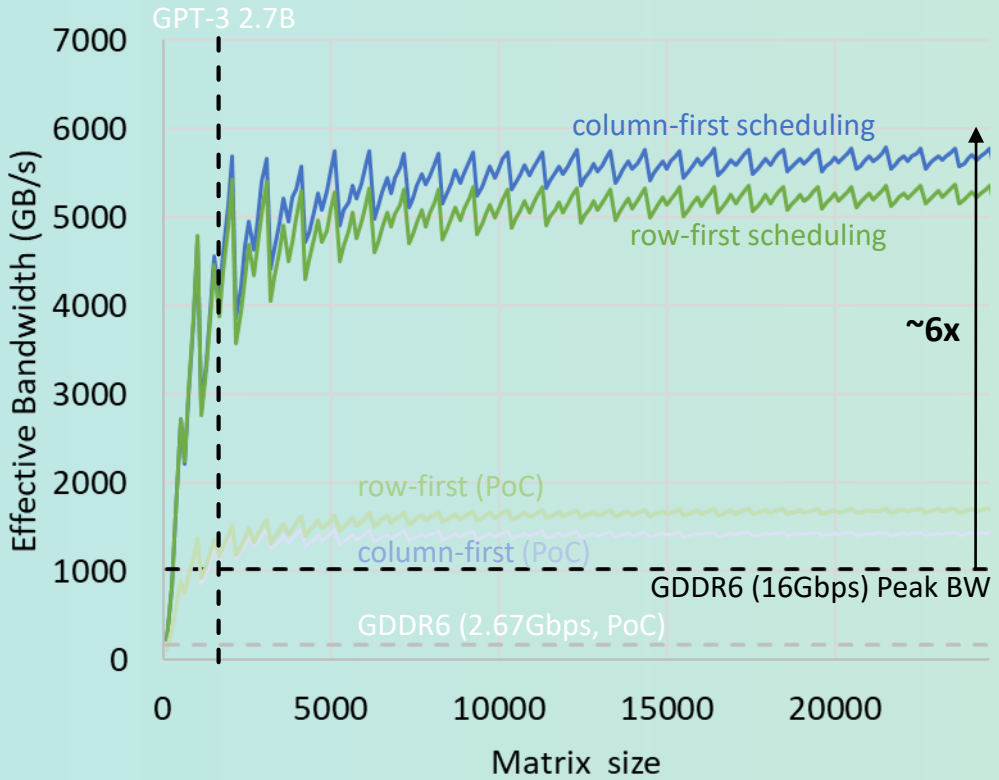
SK hynix's Accelerator-in-Memory (AiM) and AiM-centric accelerator (AiMX) architecture

AiMX System-level Performance Analysis and System Integration/Deployment Options

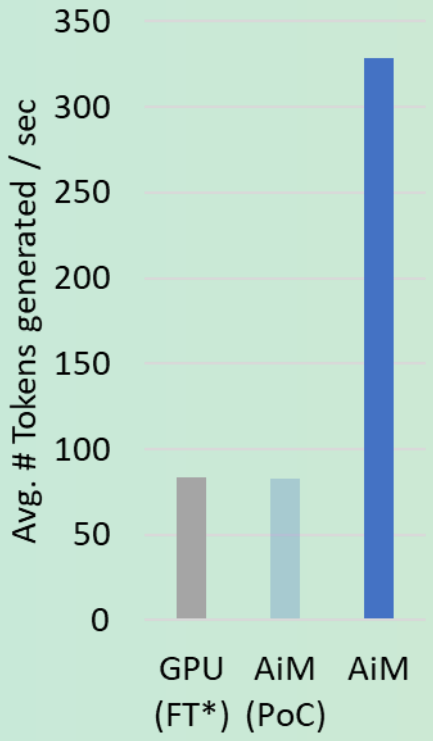
System-level Performance Analysis

- **6 TB/s of effective GEMV (minimum) memory bandwidth** of GDDR6-AiM (@16Gbps) (or 6x higher than baseline GDDR6 peak bandwidth) with optimized tiling strategies

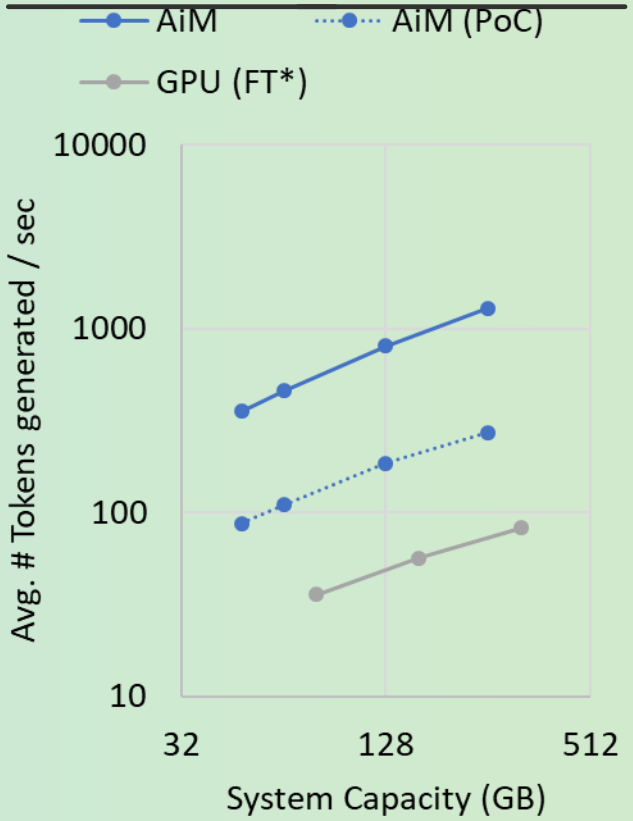
Bandwidth Achieved (GEMV)



Single Card (GPT-3 6.7B)



Multi-card Scaling (20B)

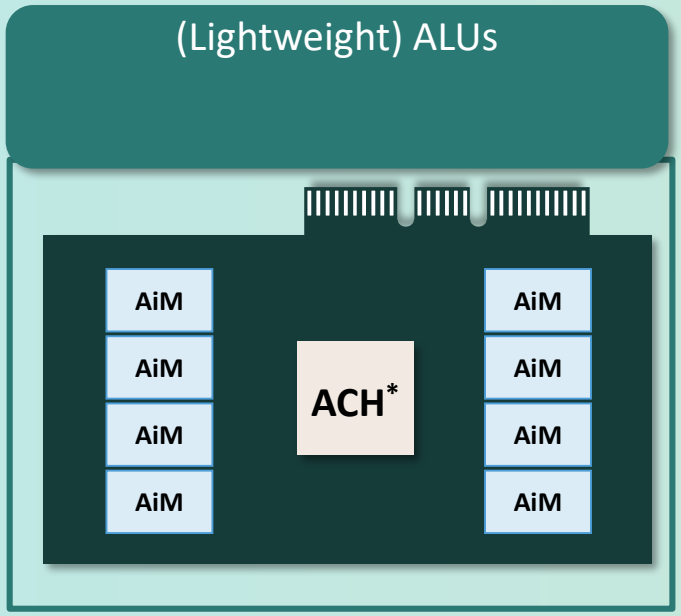


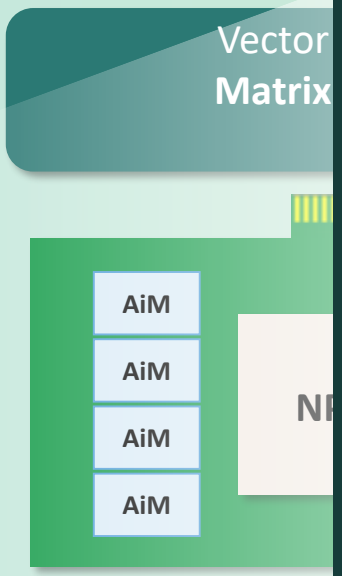
Note: Performance analysis of AiM (@16Gbps) is based on AiM cycle-accurate performance model, verified with performance measurements of AiM PoC (@2 ~ 2.67Gbps).

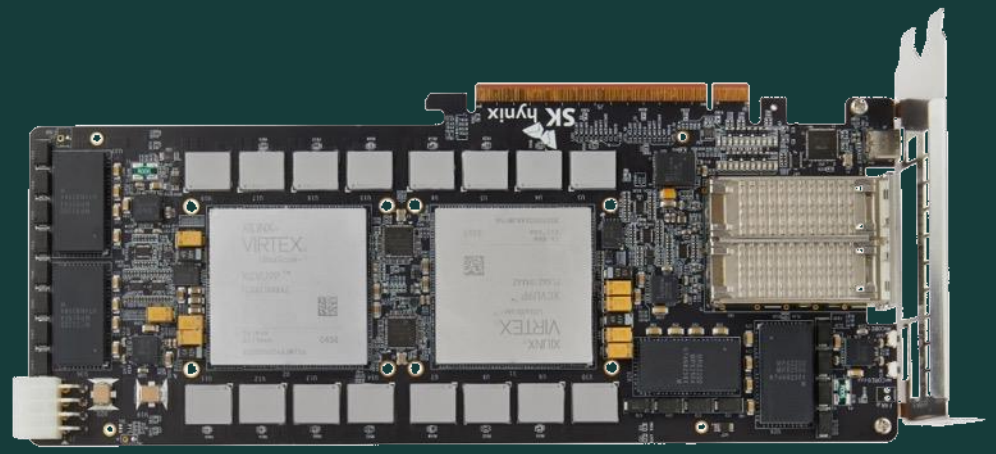
(FT) FasterTransformer: A100 GPU (SXM), batch1 performance*

System Integration Options

“ Possibilities are ENDLESS “







SK hynix

AiM-centric Accelerator Card

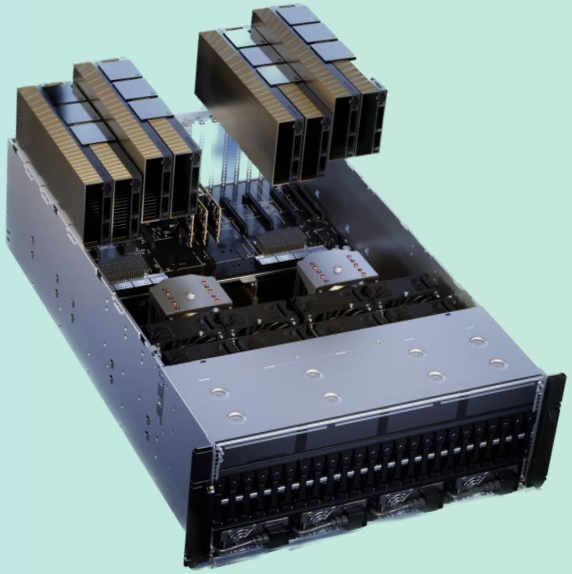
Domain Specific for LLM inference
High Performance
Cost & Energy Efficient

(*) ACH: AiM-control Hub

Example System Deployment Case: GPU + AiMX

Prompt Stage
(Question Understanding)

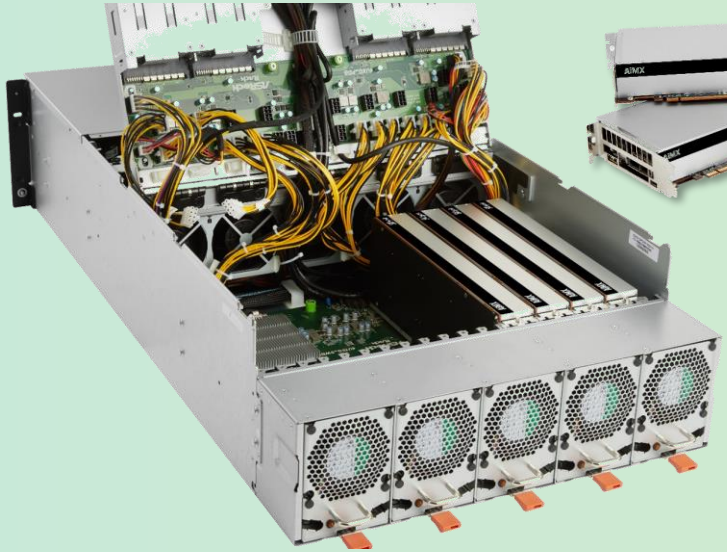
- Input tokens are processed in parallel
- Needs just 1 time model data read for all tokens
- → Computing intensive

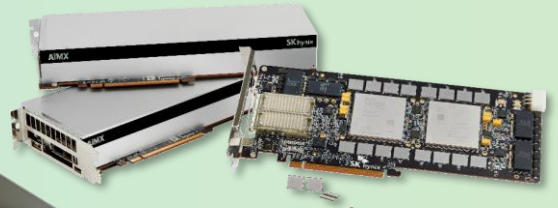


GPU system

Response Stage
(Answer Generating)

- Output tokens are processed in serial
- Needs model data read in each token generation
- → “Extremely” memory intensive





AiMX cards

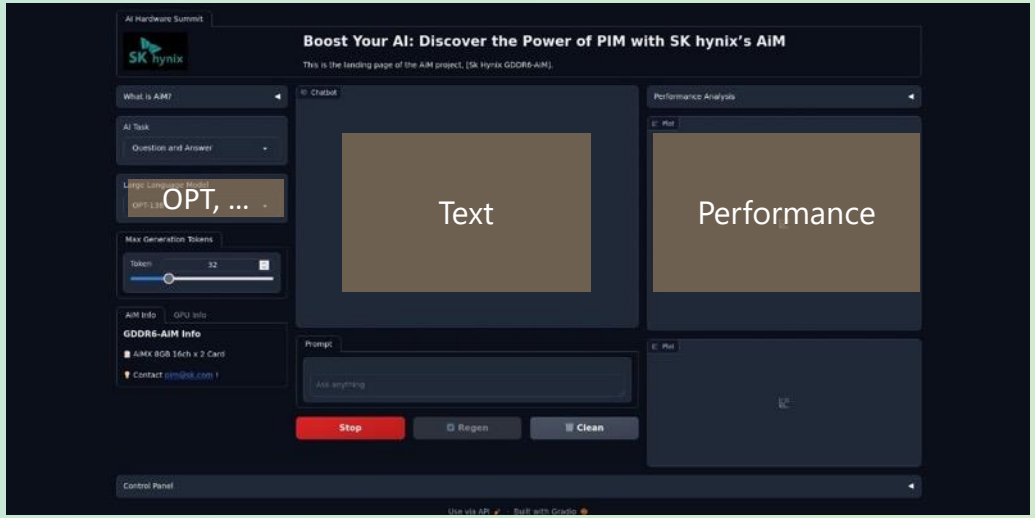
AiMX system

Showcase: Generative Q&A with AiMX System Prototype

- We developed an AiMX proto. card using FPGA chip and built an AiMX reference system optimized for LLM with GPU cards for Prompt Stage and AiMX cards for Response Stage
- Generative Q&A showcase with AiMX prototype system at SK hynix booth

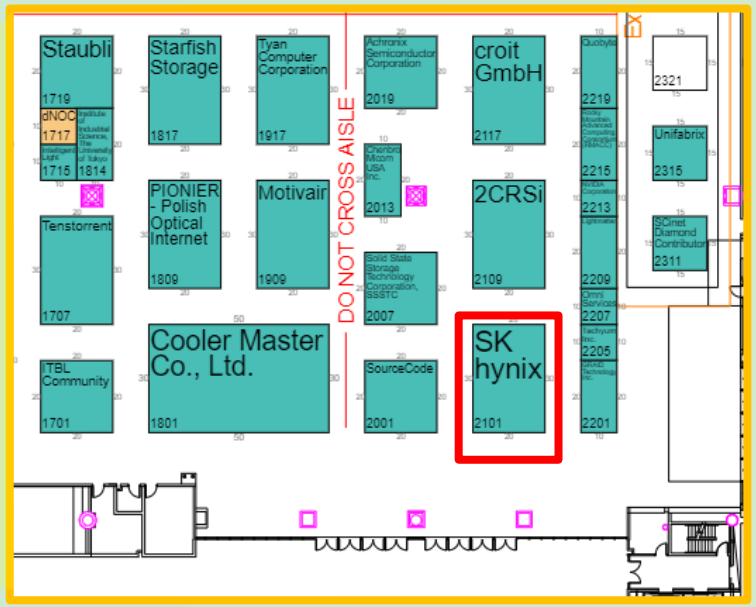
AiMX reference system

LLM inference (Q&A, Text Gen.) Demonstration GUI



Visit us at Booth #2101

SK hynix Booth : #2101




SK hynix
SK hynix
Accelerator in Memory
AiMX