PIM Is All You Need: A CXL-Enabled GPU-Free System for Large Language Model Inference

Yufeng Gu*, Alireza Khadem*, Sumanth Umesh, Ning Liang,
Xavier Servot, Onur Mutlu, Ravi Iyer, and Reetuparna Das

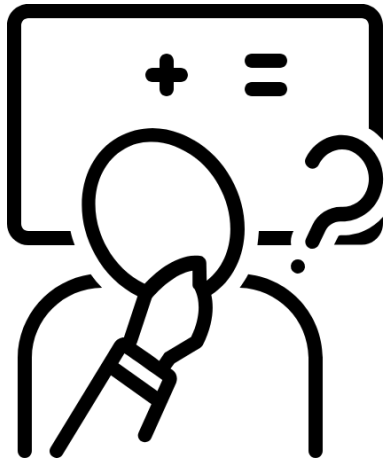
* Equal Contribution

April 2025

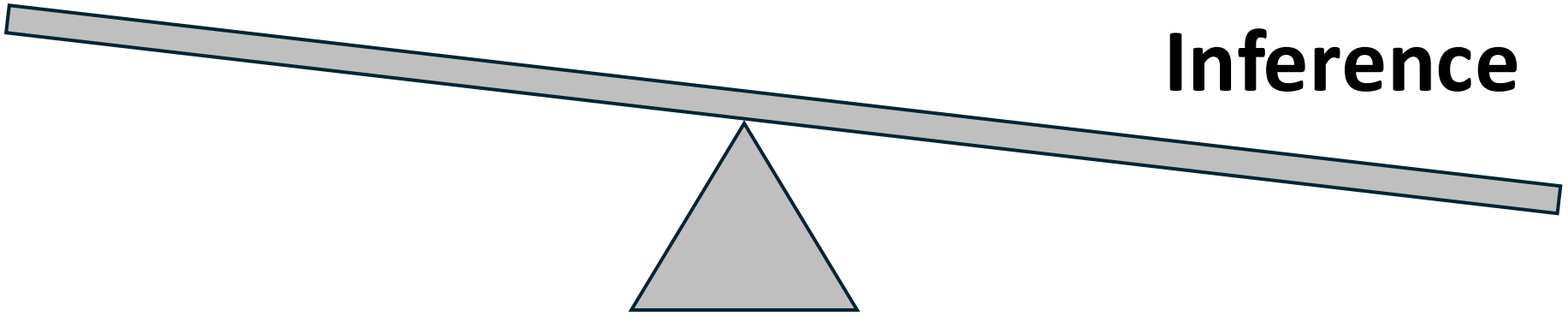
Open-source: https://github.com/Yufeng98/CENT


Training

2x
Carbon
Footprint

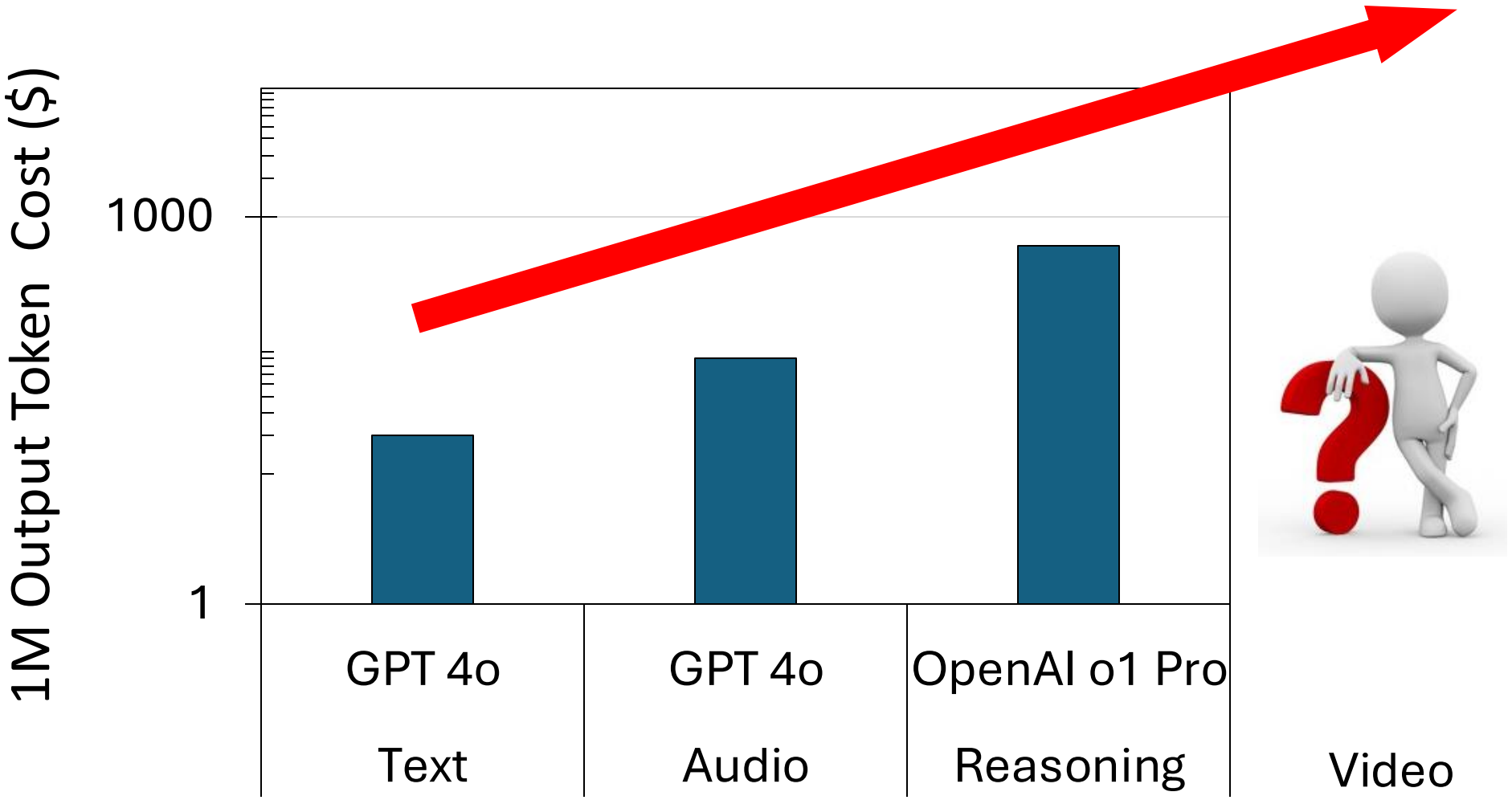


Inference



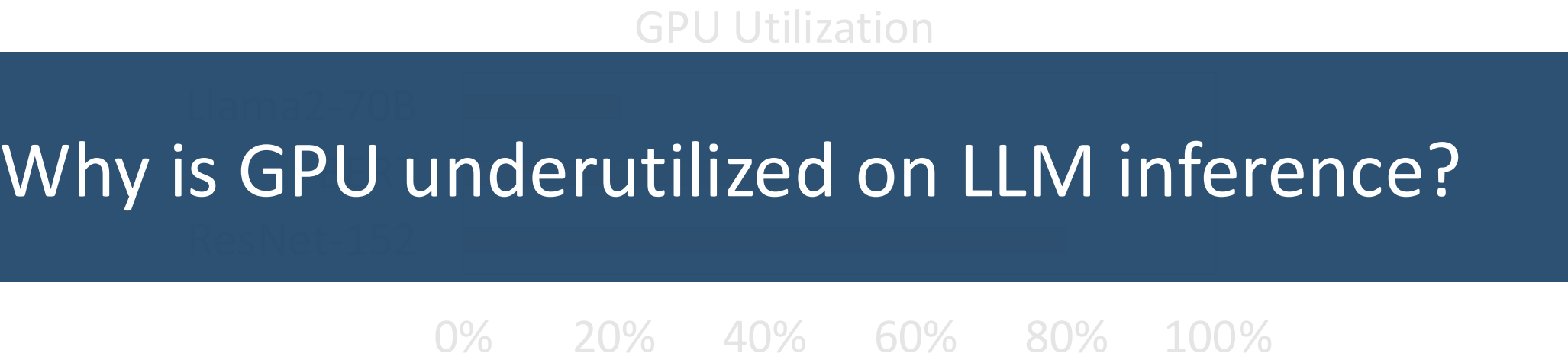
[1] Meta AI, Sustainable AI: Environmental Implications, Challenges and Opportunities, MLSys'22

LLM Inference Cost Is Increasing



[1] https://openai.com/api/pricing/

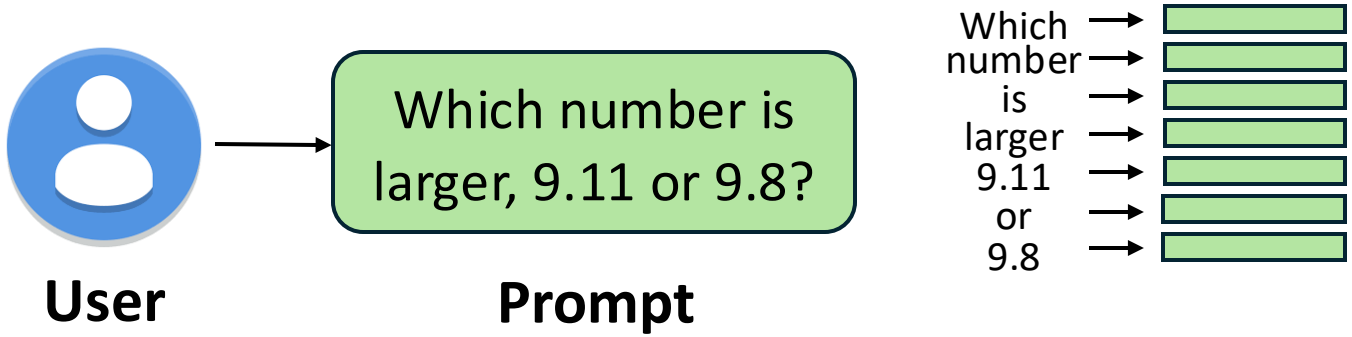
Expensive GPU Server Is under Low Utilization

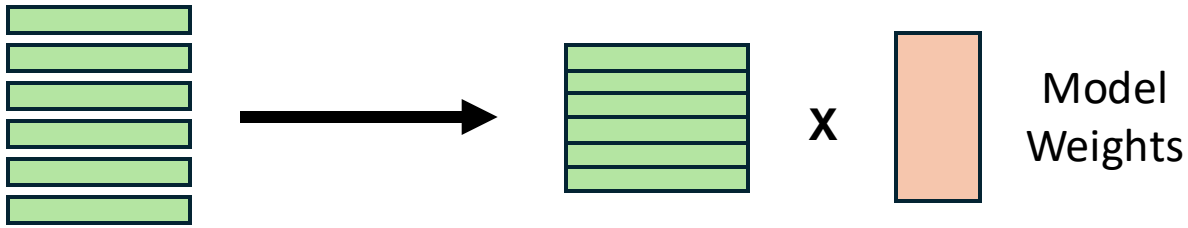


Why is GPU underutilized on LLM inference?

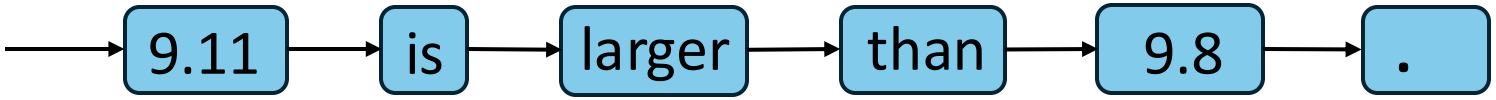
How does LLM Inference work?

Prefill Stage
(Prompt Summarization)

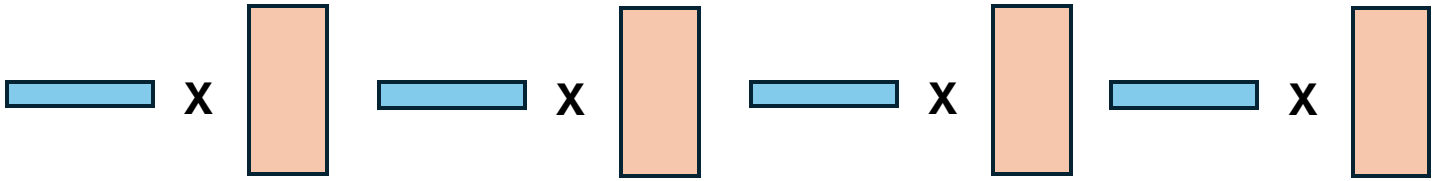




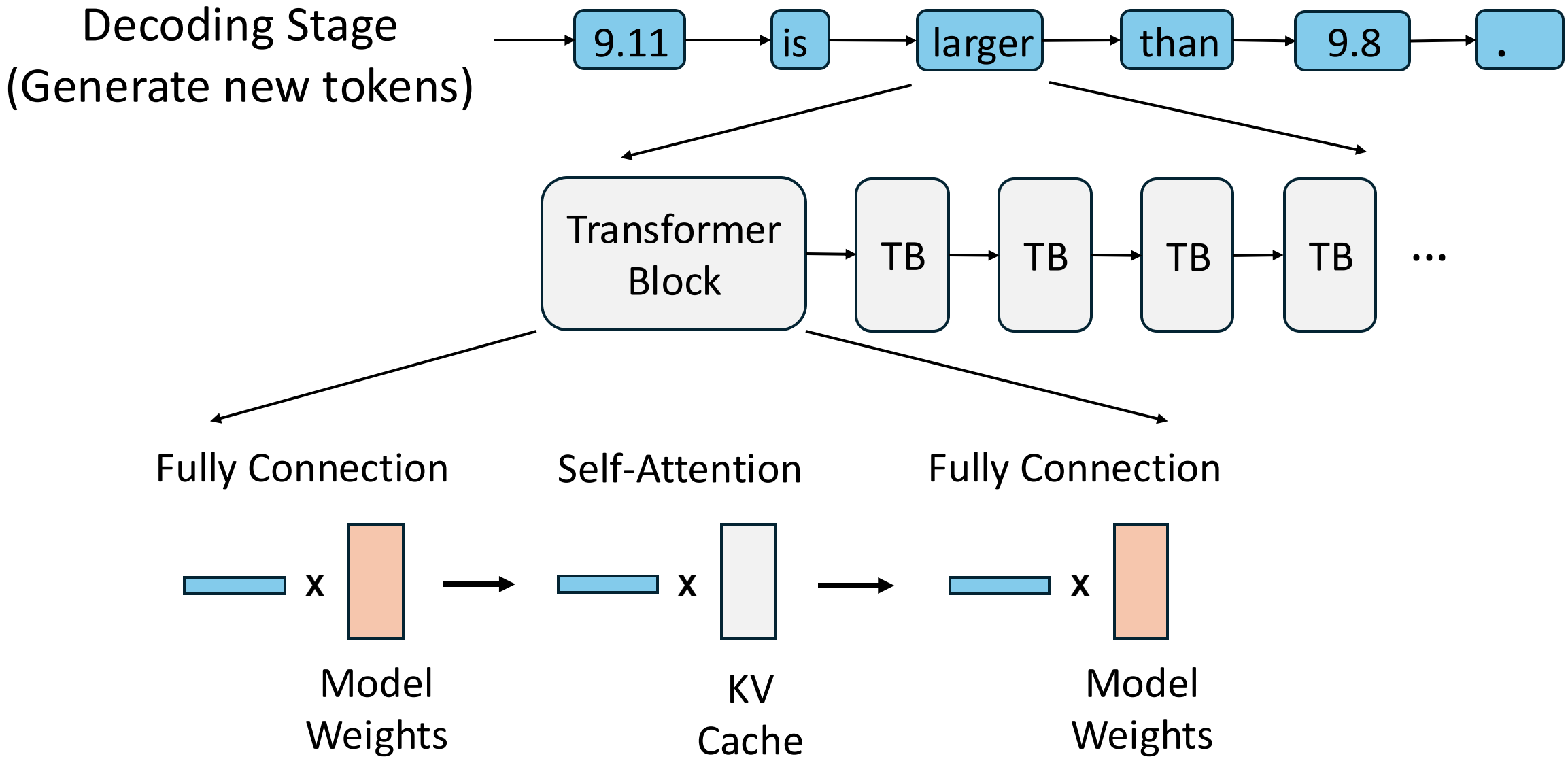
Decoding Stage
(Generate new tokens)



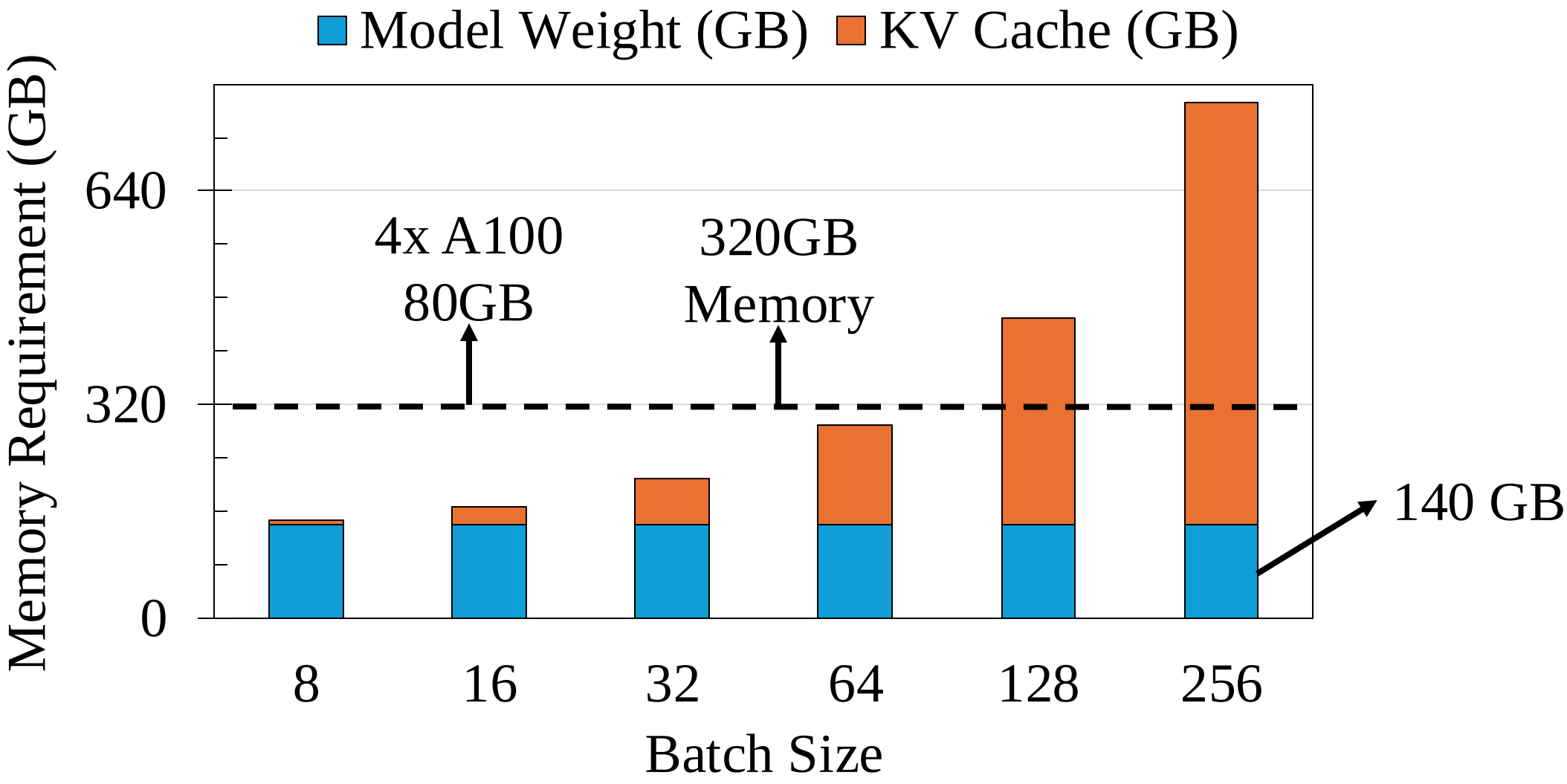
Generate output tokens



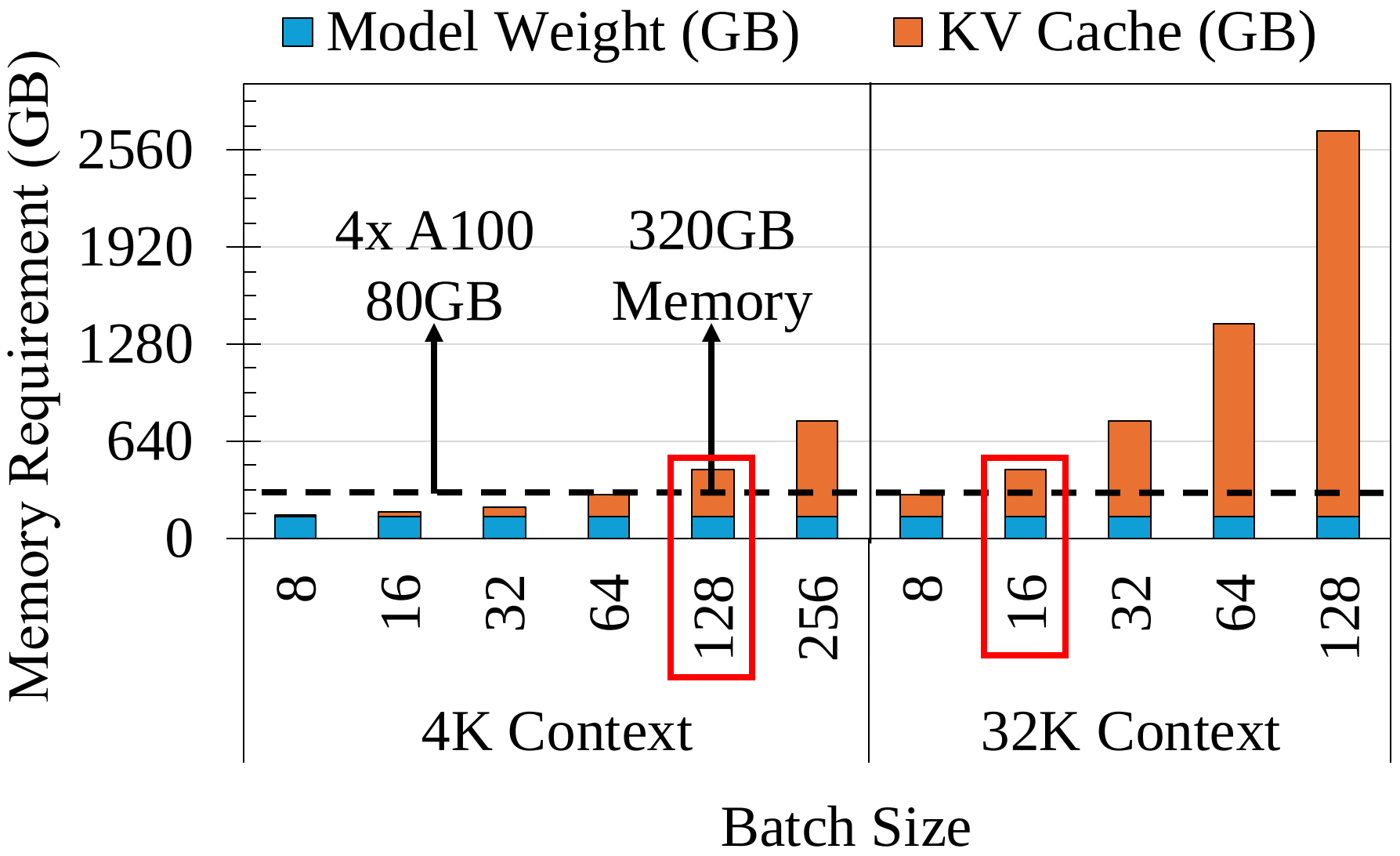
LLM Inference Decoding



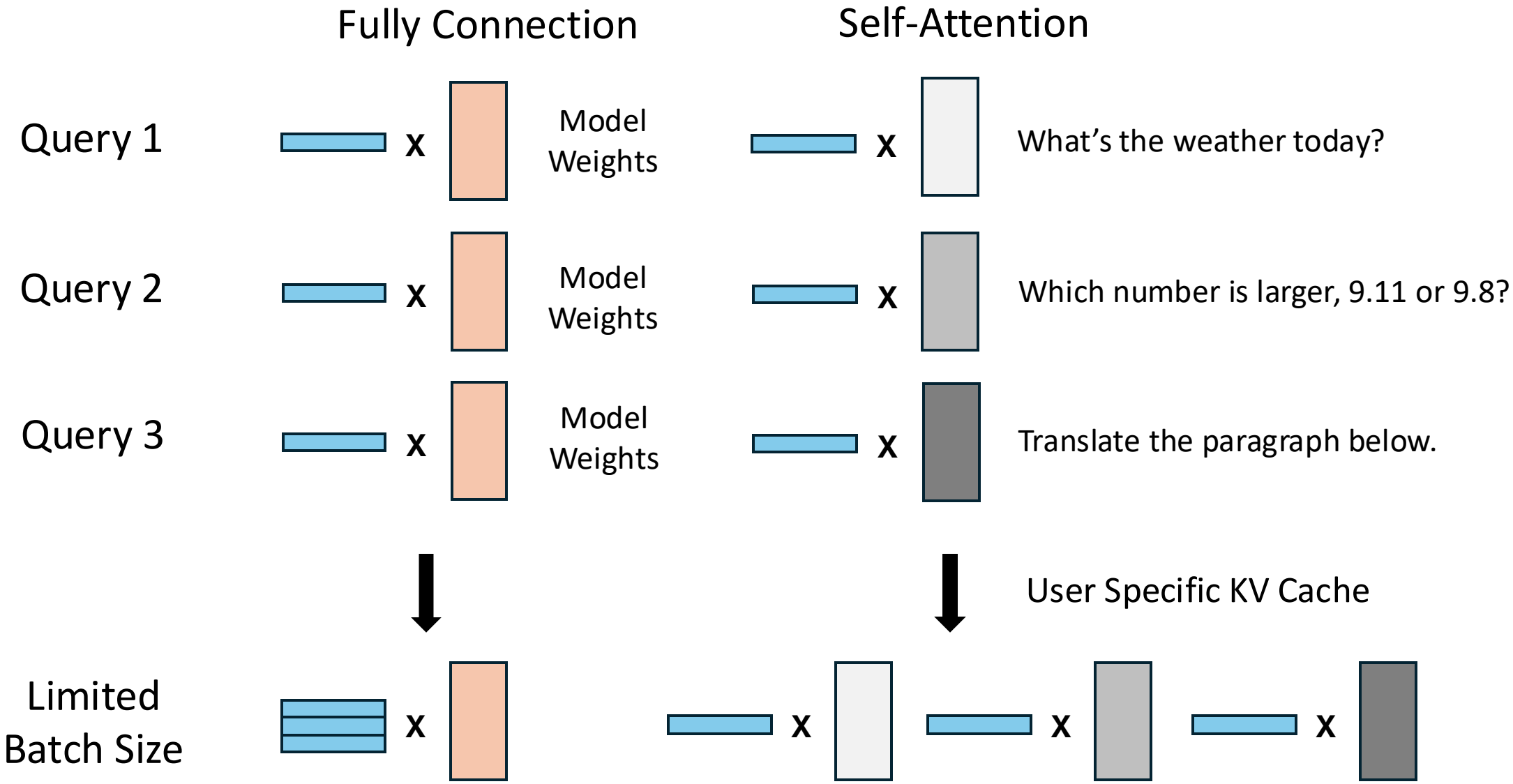
Llama 70B Inference Memory Requirement



GPU Memory Capacity Limits LLM Inference Batch Size

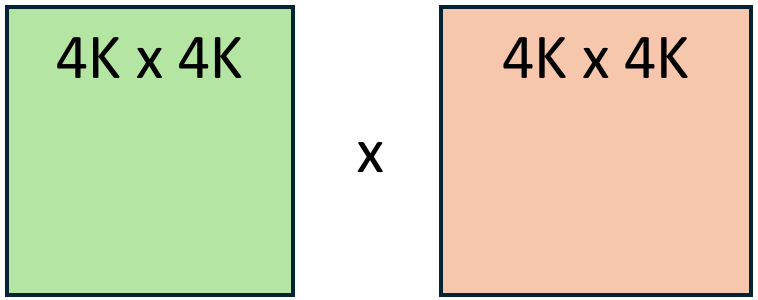


Batch Queries in Decoding

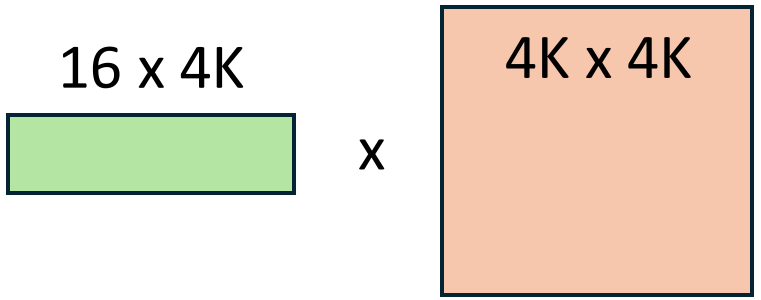


LLM Inference Has Low Operational Intensity

GEMM



LLM Inference



Operational Intensity = **2000** Ops/Byte

$$\frac{4K * 4K * 4K * 2 \text{ Operations}}{(4K * 4K + 4K * 4K) * 2 \text{ Byte}}$$

Operational Intensity = **16** Ops/Byte

$$\frac{16 * 4K * 4K * 2 \text{ Operations}}{(16 * 4K + 4K * 4K) * 2 \text{ Byte}}$$

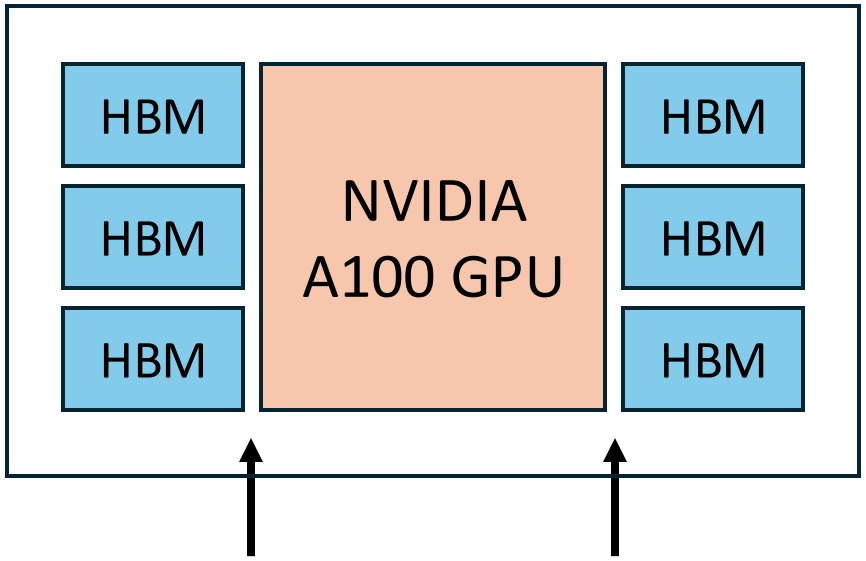
A100 GPU Operational Intensity = $\frac{312 \text{ TFLOPS (BF16)}}{2.0 \text{ TB/s Bandwidth}}$ = **156** Ops/Byte

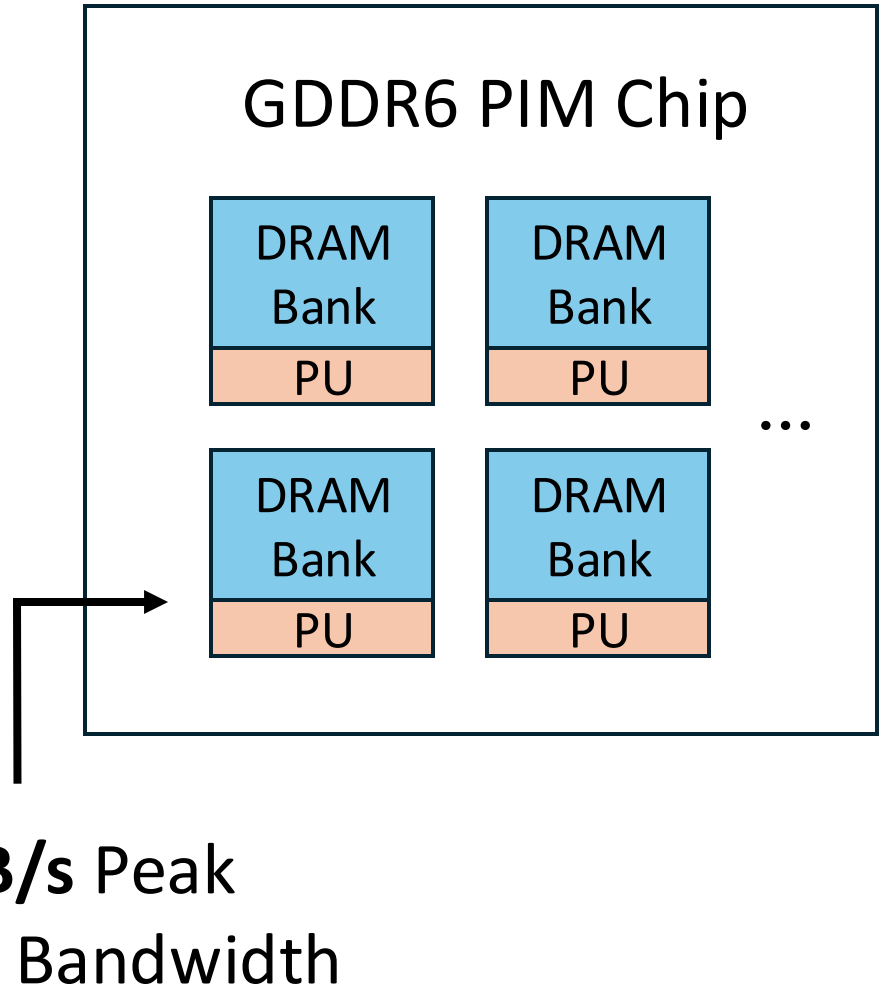
LLM Inference Has Low Operational Intensity

How can we efficiently serve LLM Inference with **Low Operational Intensity**

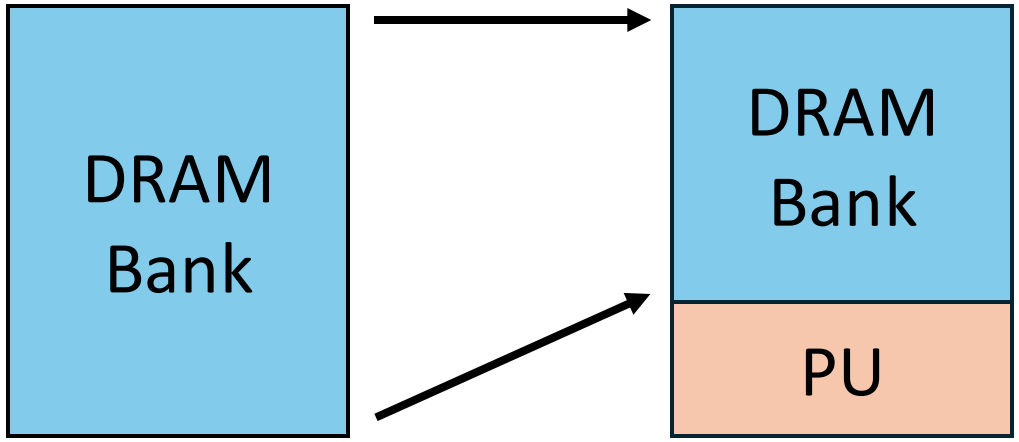
$$\text{Operational Intensity} = \frac{\text{Compute Throughput}}{\text{Memory Bandwidth}}$$

PIM Provides High Memory Bandwidth





Individual PIM Device Has Reduced Memory Density



	UPMEM [1]	AiM [2]	FIMDRAM [3]
Memory Density	25% - 50%	75%	75%

[1] The true processing in memory accelerator. In 2019 IEEE Hot Chips 31 Symposium (HCS), pages 1–24. IEEE Computer Society, 2019
[2] System architecture and software stack for GDDR6-AiM. In 2022 IEEE Hot Chips 34 Symposium (HCS), pages 1–25. IEEE, 2022.
[3] 25.4 a 20nm 6gb function-in-memory DRAM, based on HBM2 with a 1.2 tflops programmable computing unit using bank-level parallelism, for machine learning applications. In 2021 IEEE International Solid-State Circuits Conference (ISSCC), volume 64, pages 350–352. IEEE, 2021.

LLM Inference Requires Large Memory Capacity

Llama 70B
4K Context
Batch = 128

450 GB

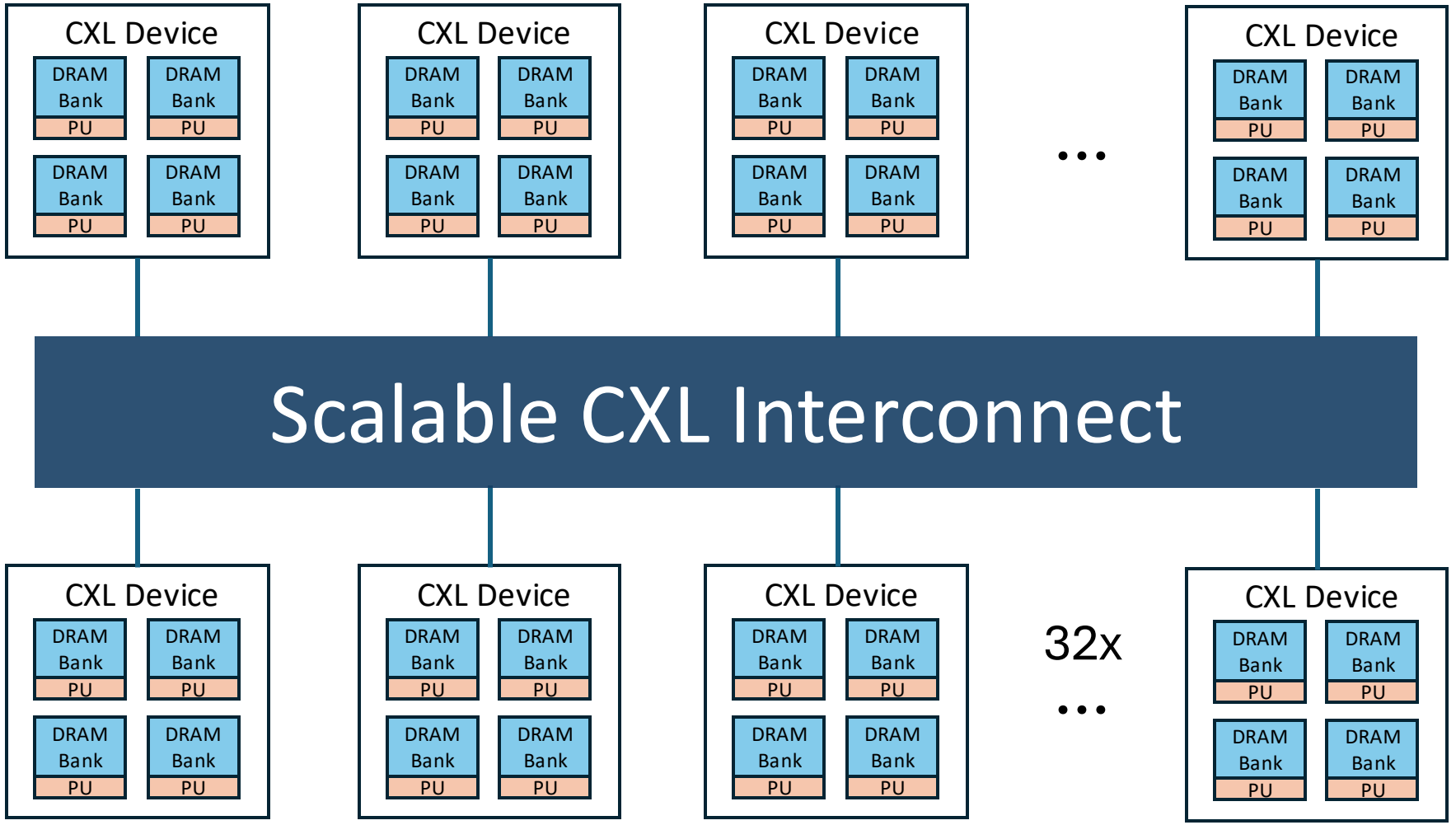
140 GB
Model Weights

310 GB
KV Cache

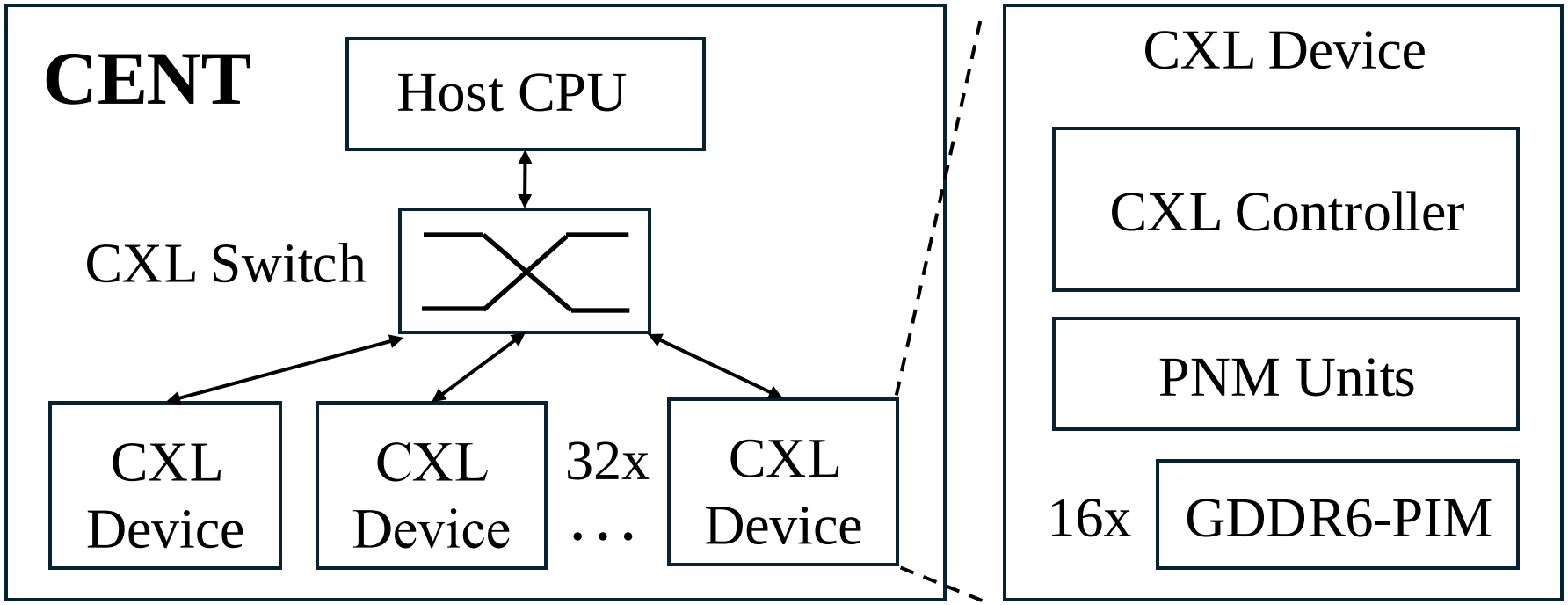
	Llama2	Llama3	DeepSeek R1	OpenAI o1	Gemini 1.5 Pro	Grok-3	Claude 3.5
Model	70 B	405 B	671 B	-	-	-	-
Contexts	4K	128 K	128 K	200 K	2 M	1 M	200 K

https://platform.openai.com/docs/models#o1, https://developers.googleblog.com/en/new-features-for-the-gemini-api-and-google-ai-studio/, https://x.ai/blog/grok-3, https://docs.anthropic.com/en/docs/about-claude/models, The Llama 3 Herd of Models, DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

CXL Interconnect Provides Substantial Memory Capacity



CENT Architecture



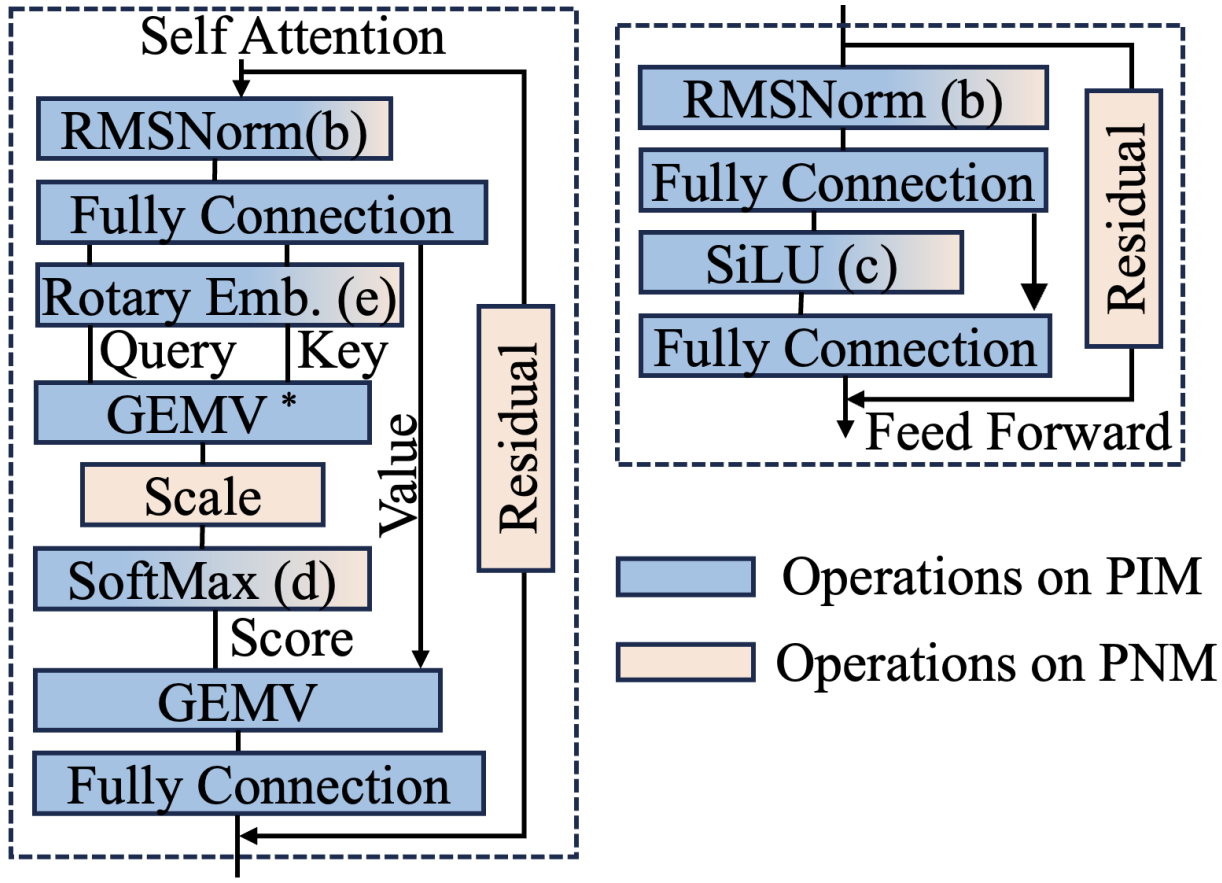
Full Transformer Block Execution on the CXL Device

PIM Module **99%**

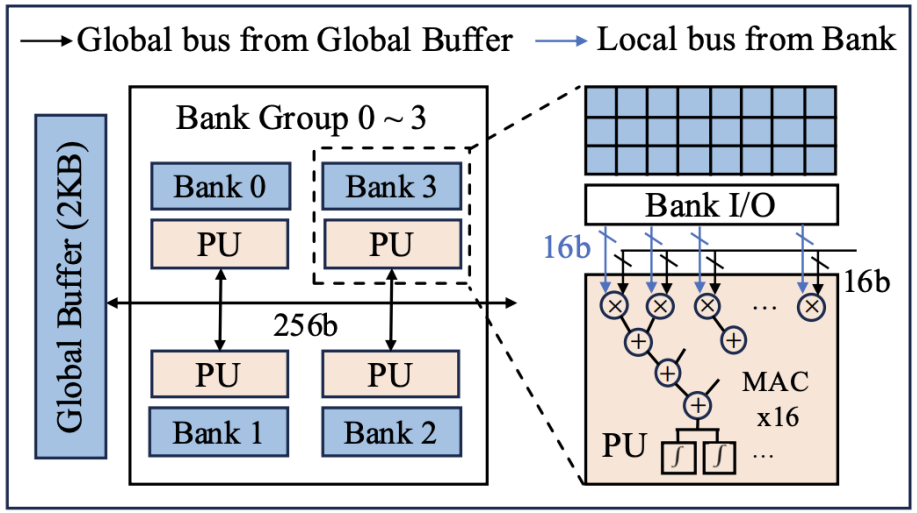
- Vector Multiplication
- Element-wise Multiplication

PNM Accelerator

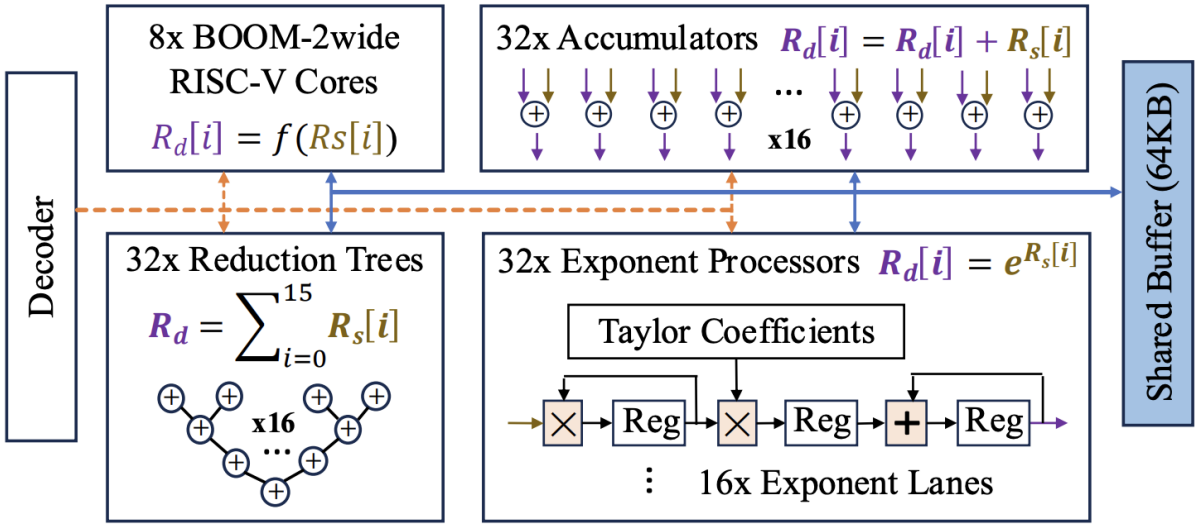
- Vector Addition Units
- Reduction Trees
- Exponent Processors
- RISC-V cores for Special Functions



Hierarchical PIM-PNM Design



(a) GDDR6-PIM Channel

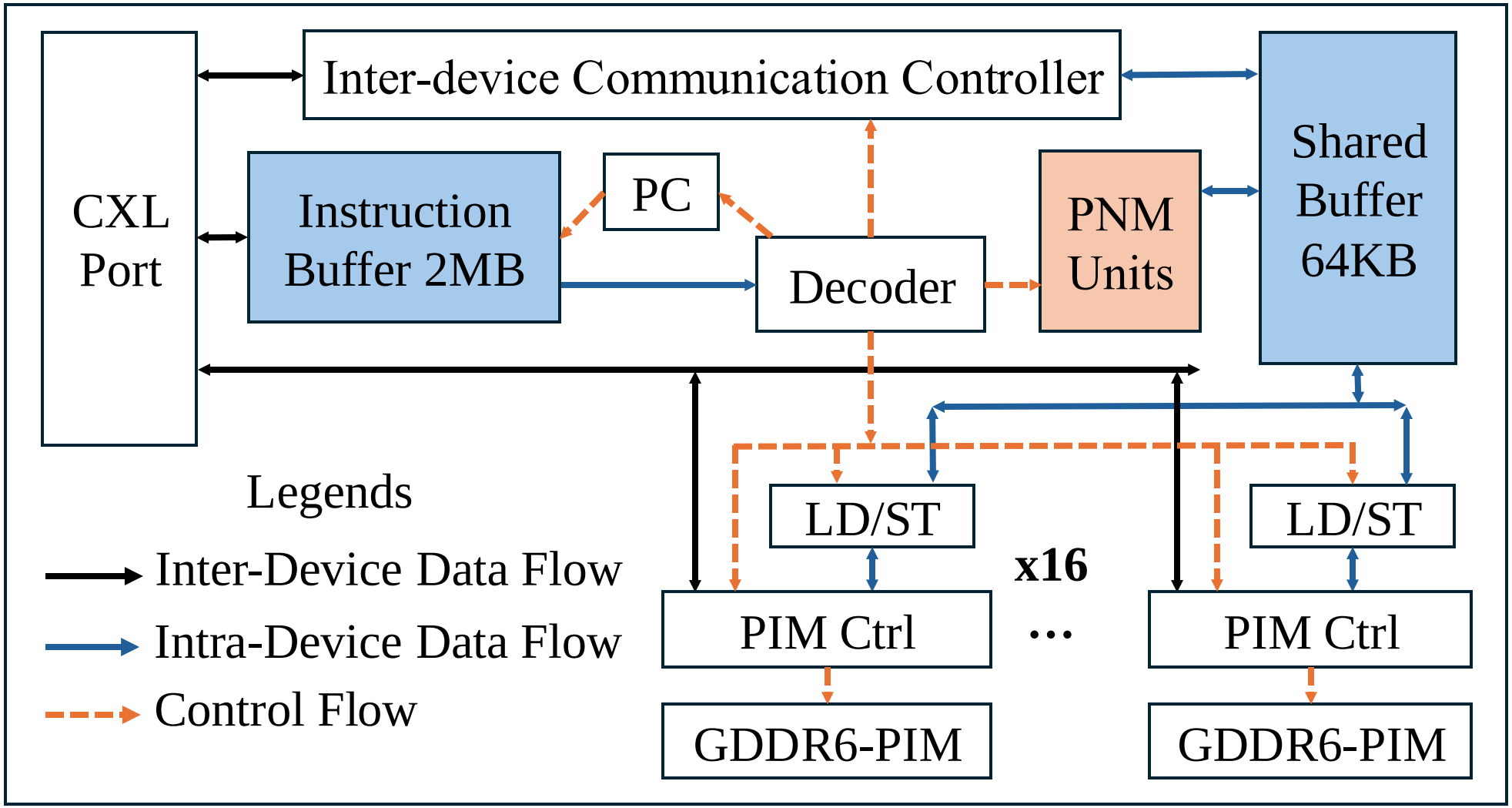


(b) PNM Units

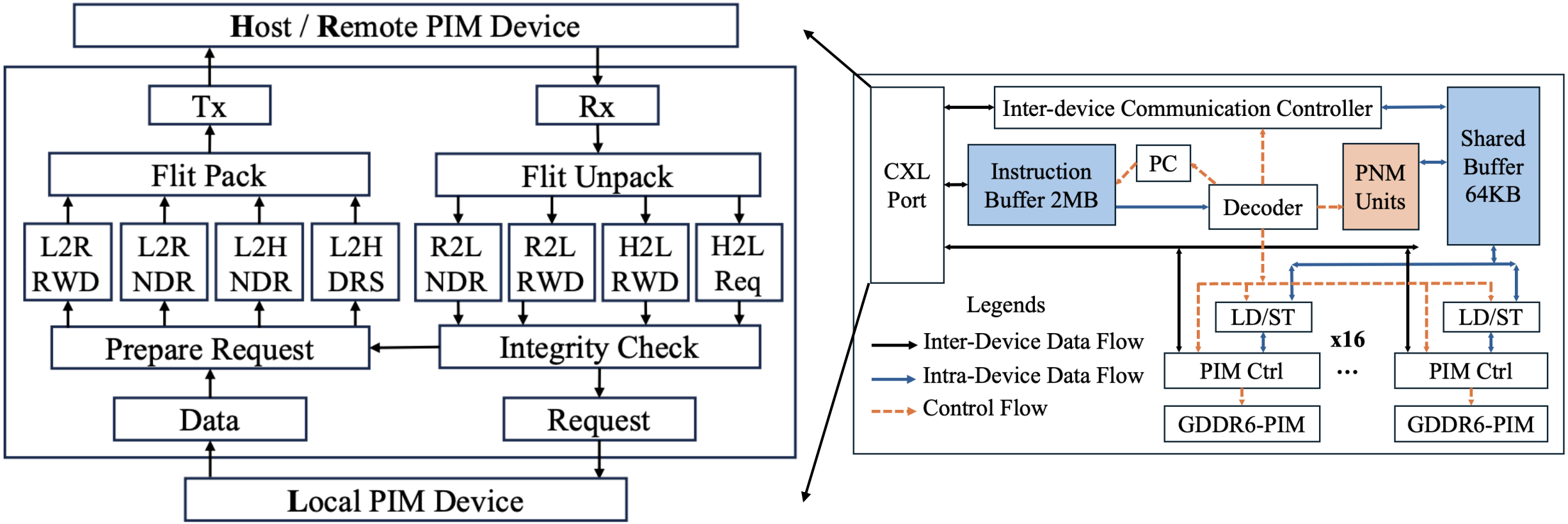
- Vector Multiplication
- Element-wise Multiplication

- Vector Addition Units
- Reduction Trees
- Exponent Processors
- RISC-V cores for Special Functions

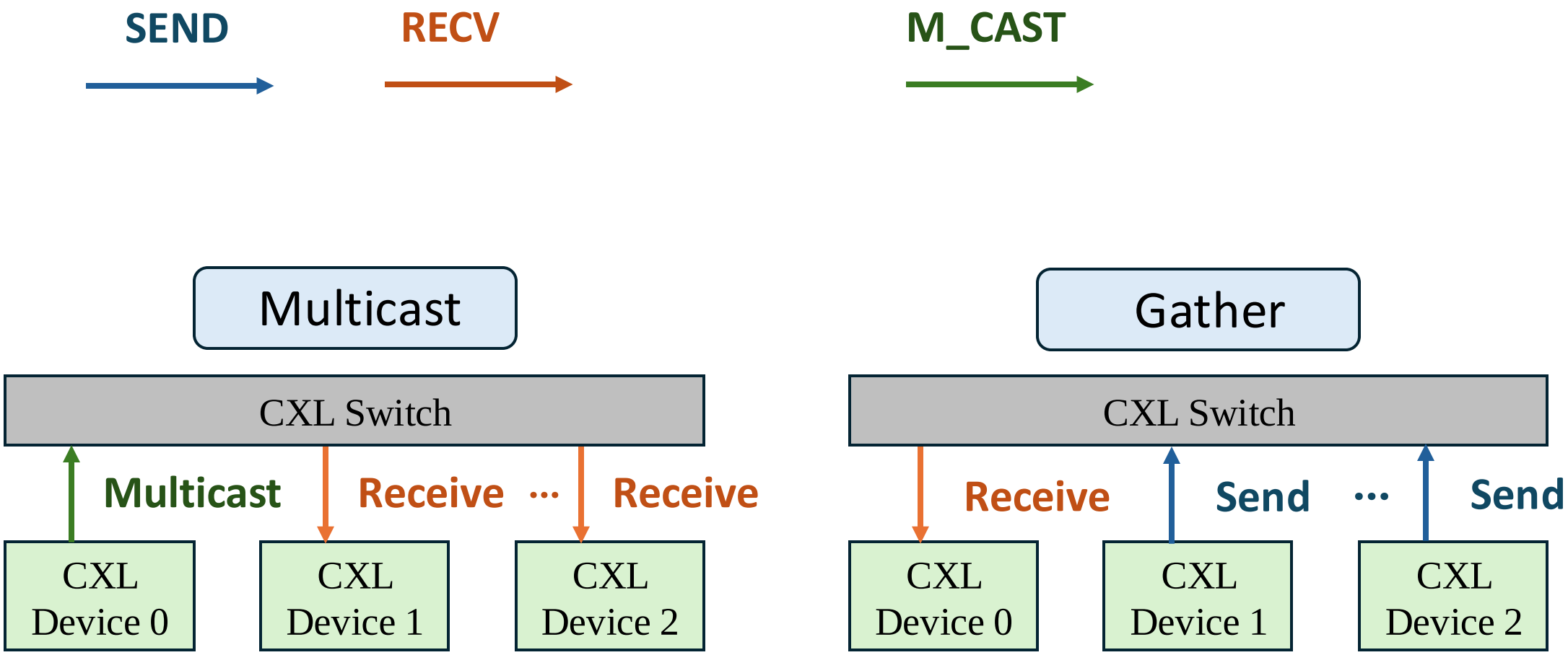
CXL Device Architecture



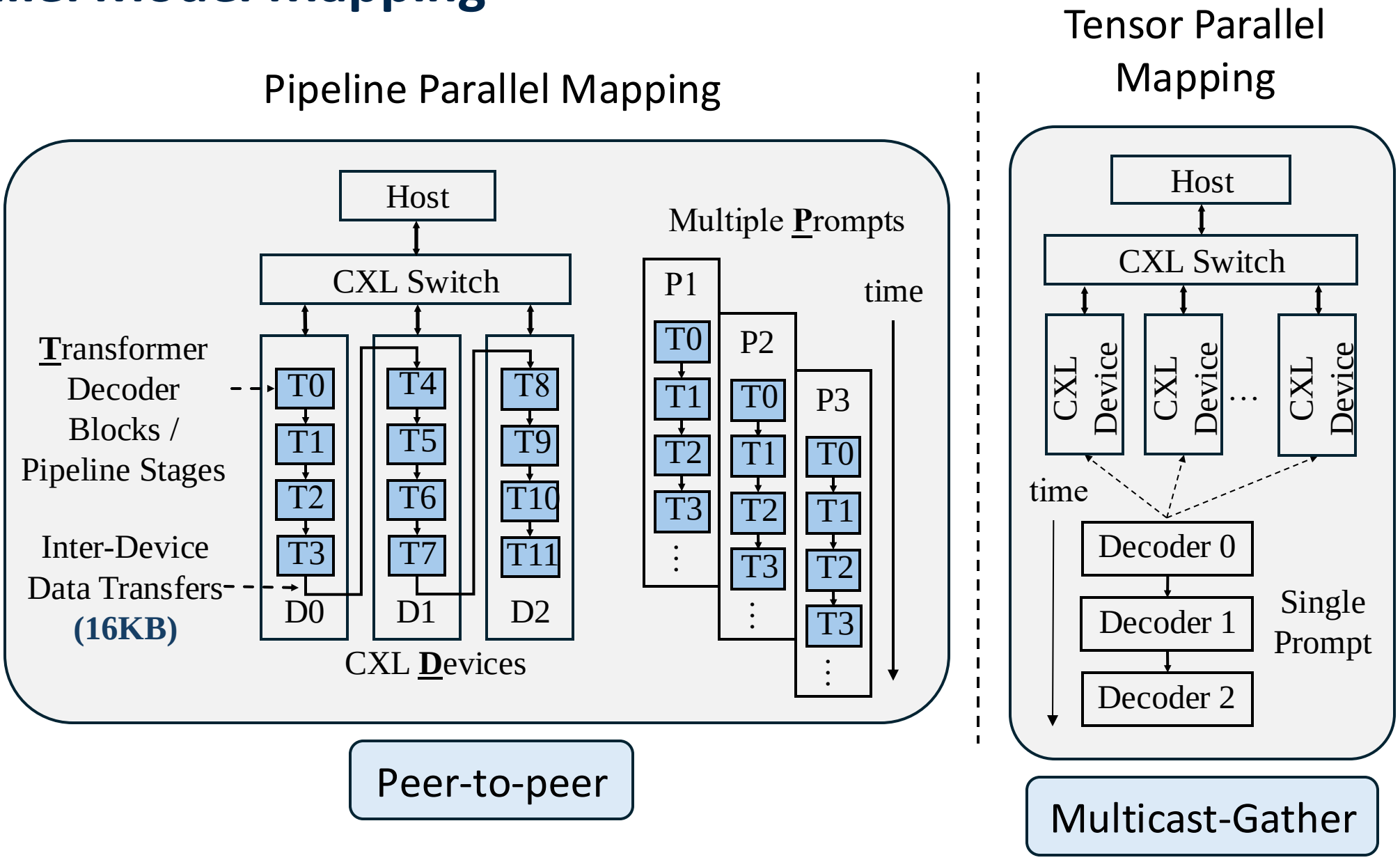
CXL Port Architecture



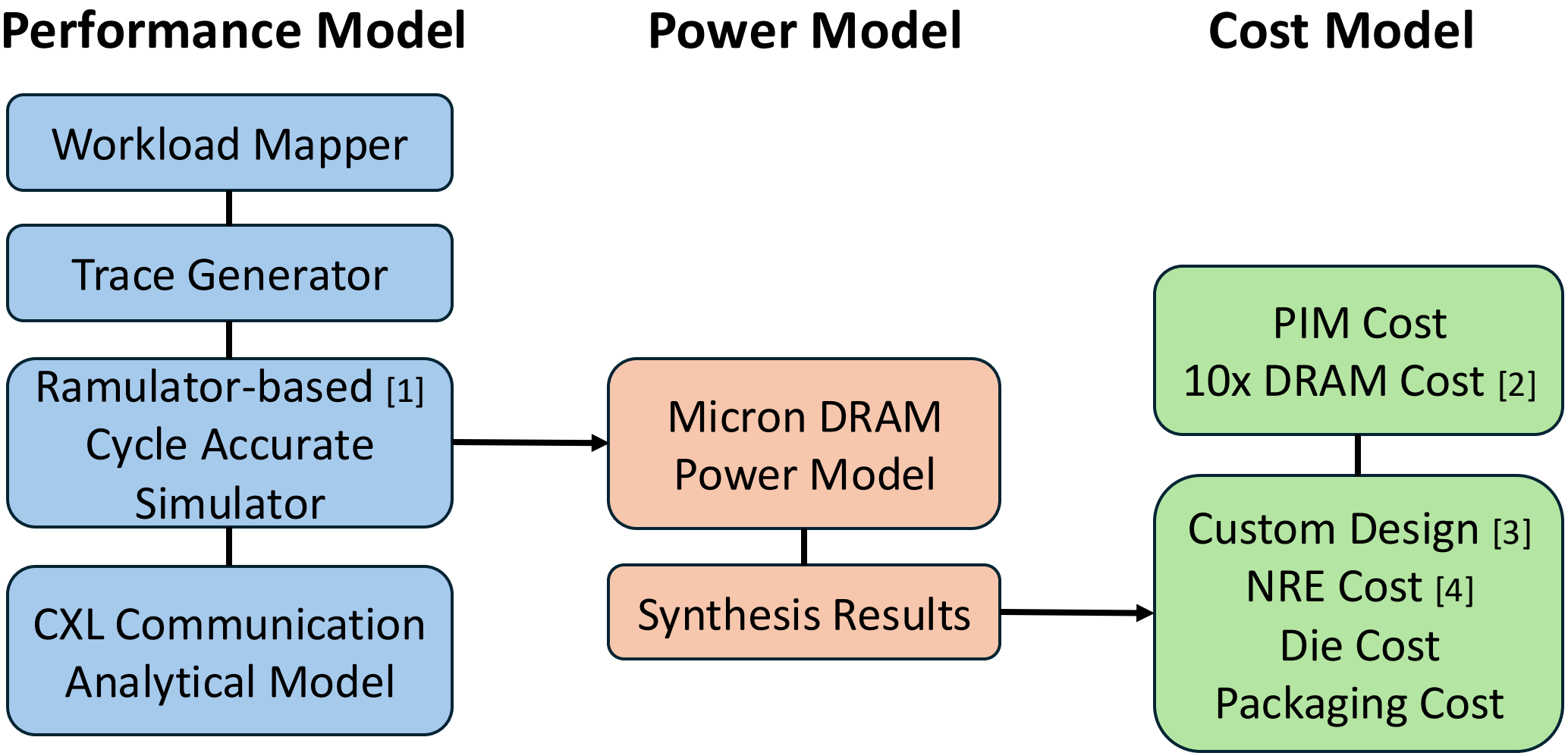
Inter-Device CXL Communication



Parallel Model Mapping



Evaluation Methodology



[1] Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator.
[2] UPMEM. Accelerating compute by cramming it into dram memory
[3] Supply chain aware computer architecture. In Proceedings of the 50th Annual International Symposium on Computer Architecture, pages 1–15, 2023.
[4] Moonwalk: Nre optimization in ASIC clouds. ACM SIGARCH Computer Architecture News, 45(1):511–526, 2017

CENT versus GPU Baseline

System	CENT	GPU
Hardware	32 CXL Devices	4 NVIDIA A100
Memory	512 GB, GDDR6	320 GB, HBM2E

CENT versus GPU Baseline

System	CENT	GPU
Hardware	32 CXL Devices	4 NVIDIA A100
Memory	512 GB, GDDR6	320 GB, HBM2E
Compute Throughput	608 TFLOPS	1248 TFLOPS

CENT versus GPU Baseline

System	CENT	GPU
Hardware	32 CXL Devices	4 NVIDIA A100
Memory	512 GB, GDDR6	320 GB, HBM2E
Compute Throughput	608 TFLOPS	1248 TFLOPS
Peak Bandwidth	**512 TB/s**	**8 TB/s**

CENT versus GPU Baseline

System	CENT	GPU
Hardware	32 CXL Devices	4 NVIDIA A100
Memory	512 GB, GDDR6	320 GB, HBM2E
Compute Throughput	608 TFLOPS	1248 TFLOPS
Peak Bandwidth	**512 TB/s**	**8 TB/s**
3-Year *Owned* Total Cost of Ownership	0.73 $/hour	1.76 $/hour
3-Year *Rental* Total Cost of Ownership	1.05 $/hour	5.45 $/hour

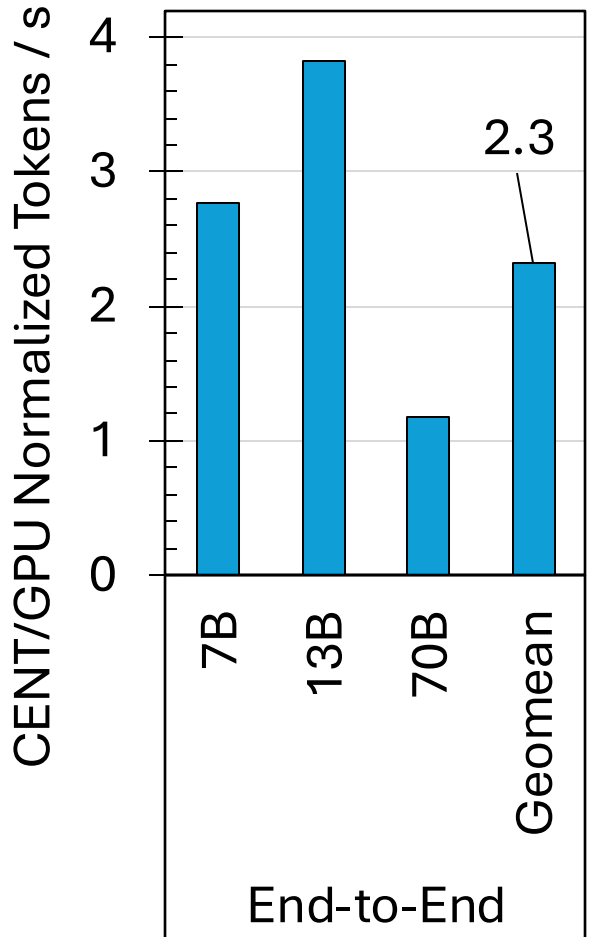
2.4x 

Cost Efficiency Comparison

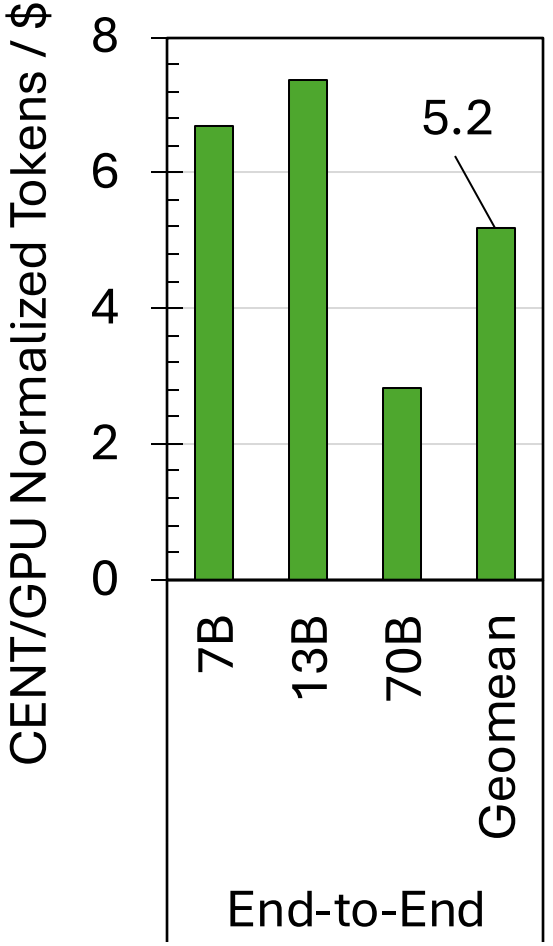
Prefill Tokens = 512
Decoding Tokens = 3.5K

	3-Year Owned TCO
GPU	1.76 $/hour
CENT	0.73 $/hour

2.4x
Hardware Cost

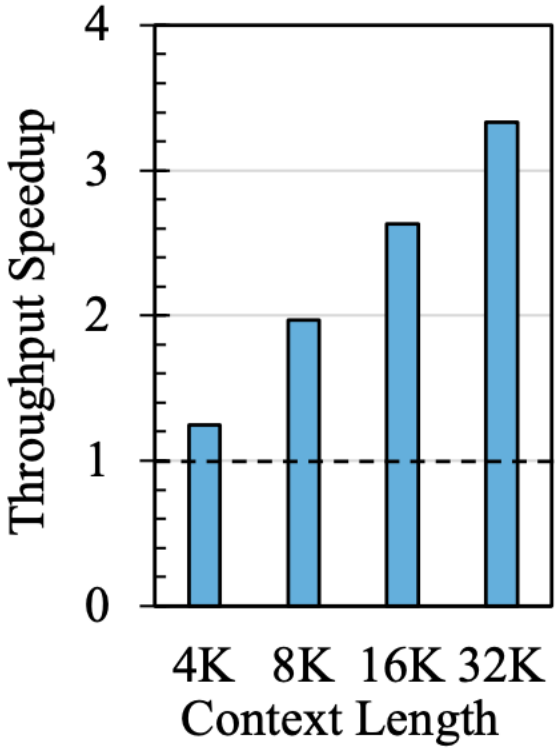


2.3x
Throughput

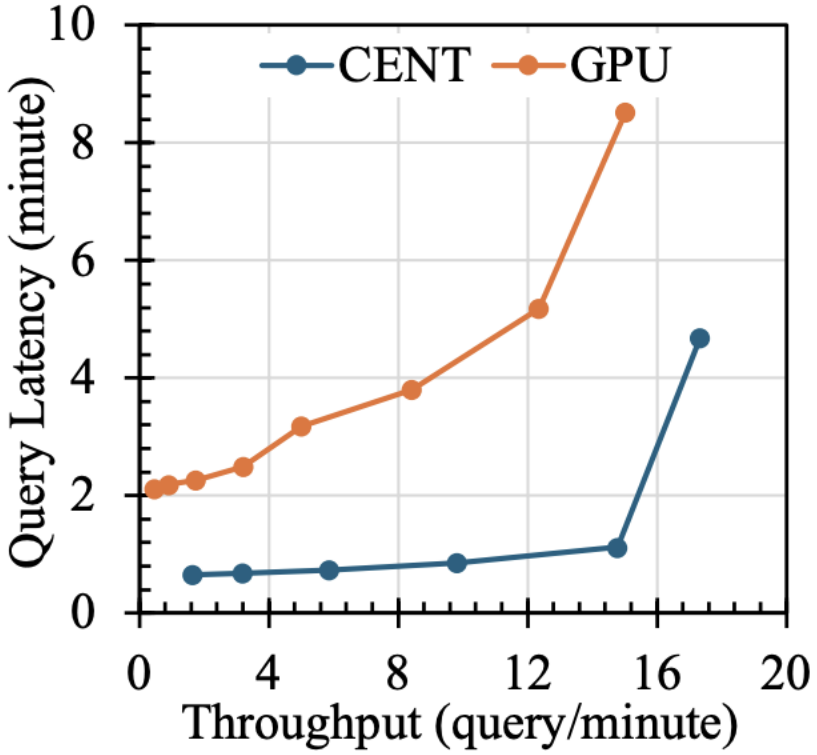


5.2x
Tokens per $

Long Context and QoS Analysis

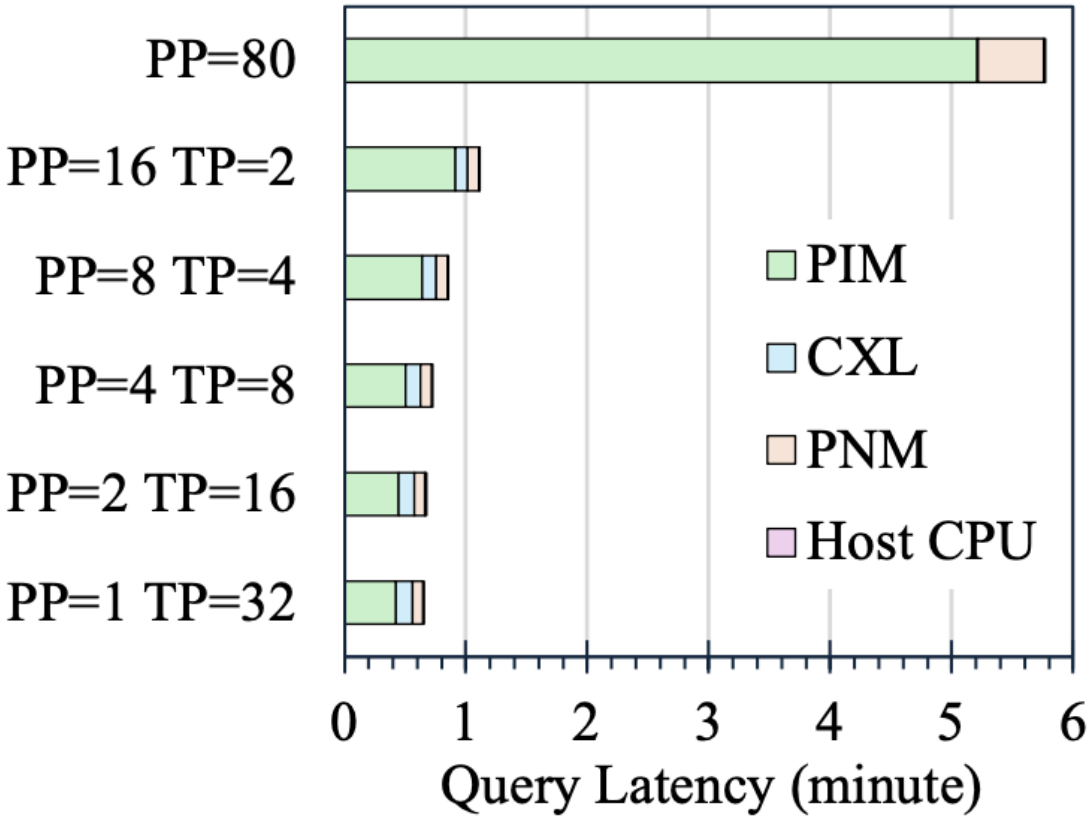


Decoding Throughput

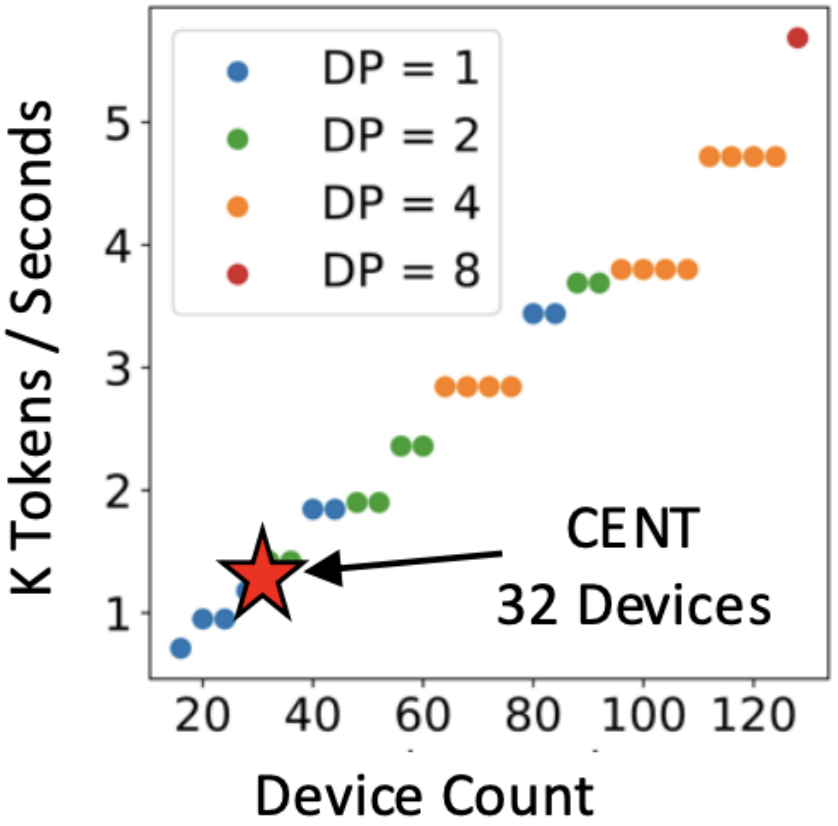


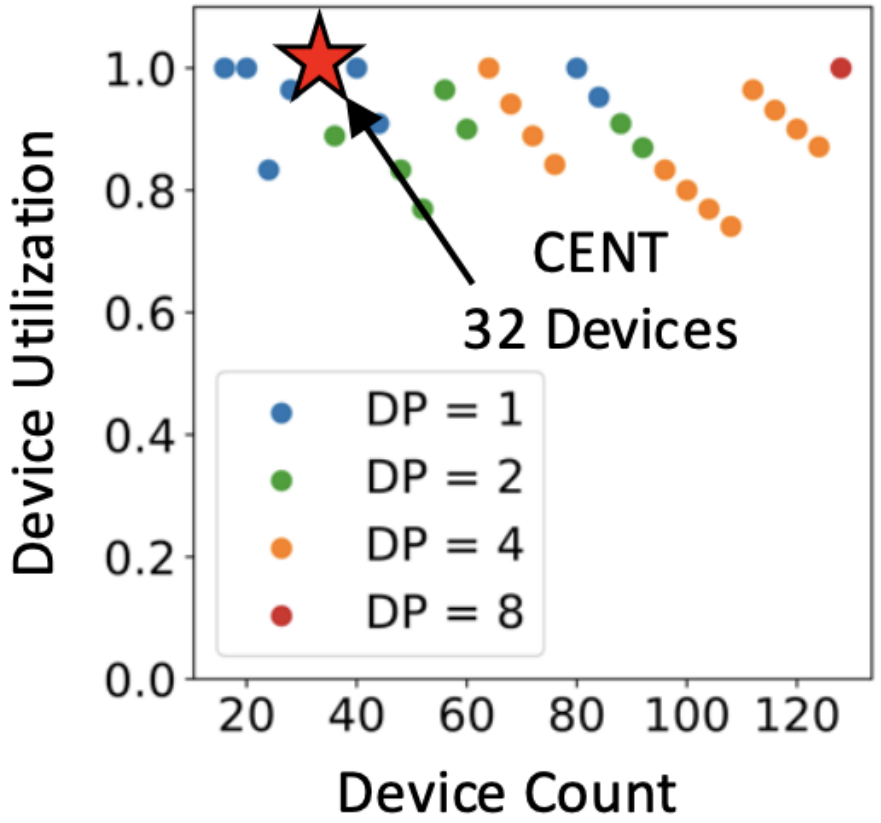
Comparison at 4K Context

CENT Latency Breakdown

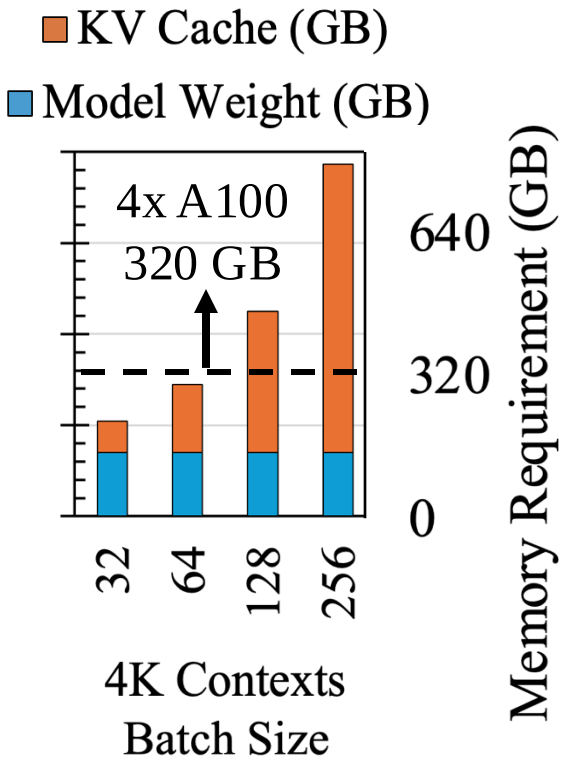


Scalability Study





CENT: A CXL-Enabled PIM System



Limited Batch

Sequential self-attention execution

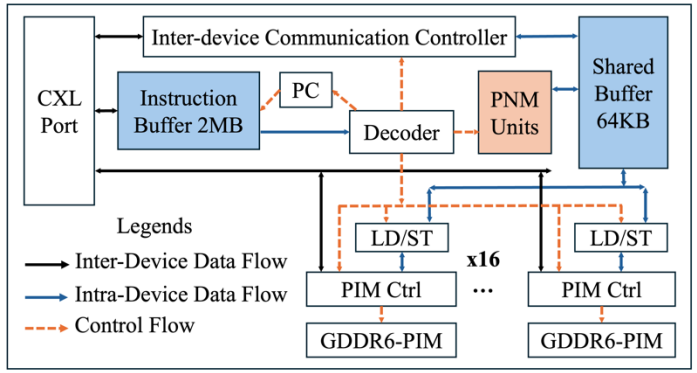
Low GPU Utilization

Low Operational Intensity

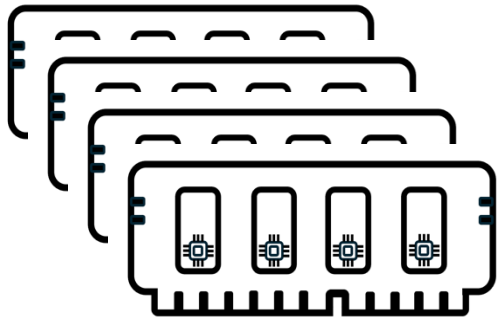
16 TB/s per GDDR6 channel

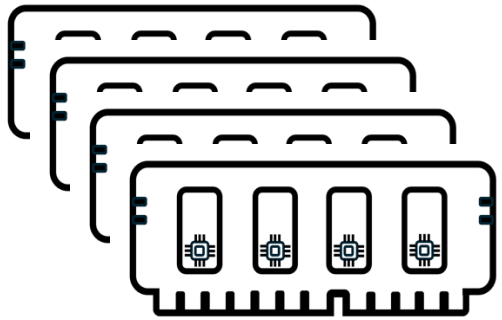
DRAM Bank

PU

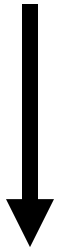


CENT: A CXL-Enabled PIM System



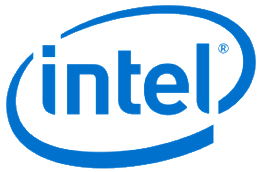


2.4x
Hardware Cost



2.3x
Throughput

5.2x
Tokens per $



PIM Is All You Need: A CXL-Enabled GPU-Free System for Large Language Model Inference

CENT Paper

CENT Artifact