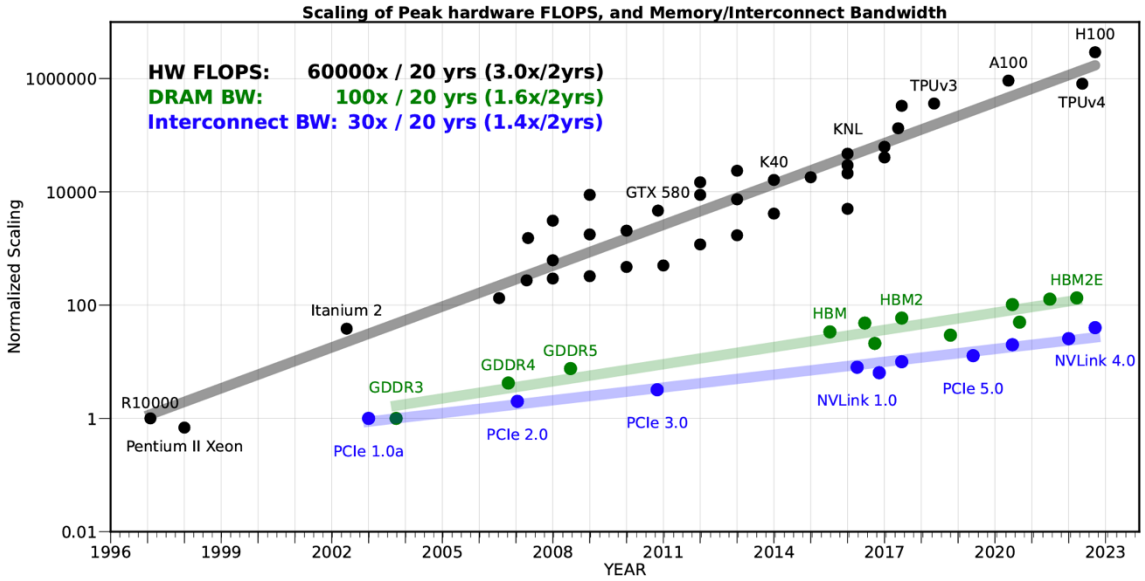
LLM Inference Performance on Chiplet-based Architectures and Systems

Surim Oh[1], Eric Qin[2], Yang Yang[2], Mengchi Zhang[2], Raj Parihar[2], Ashish Pandya[2]

[1]University of California Santa Cruz
[2]Meta

[1]Work done as an intern at Meta

“Memory Wall” in Large Language Models



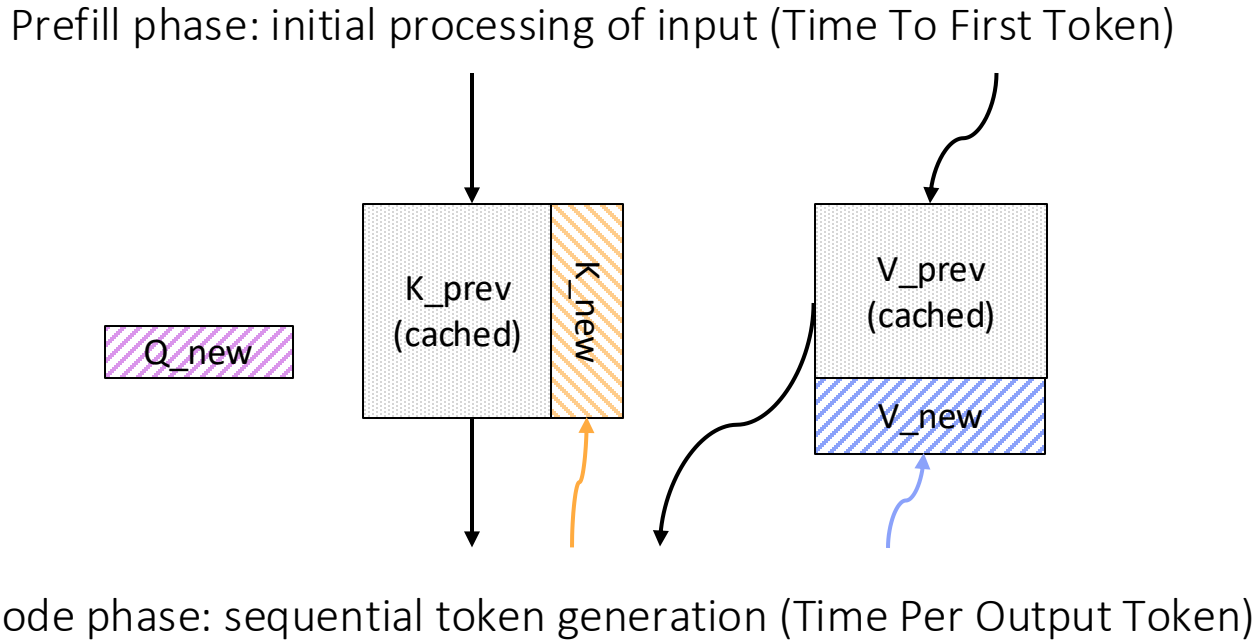
* Image from [A. Gholami et al., “AI and Memory Wall” in IEEE Micro]

Why is LLM Inference Memory Bound?

Prefill phase: initial processing of input (Time To First Token)

Decode phase: sequential token generation (Time Per Output Token)

Why is LLM Inference Memory Bound?

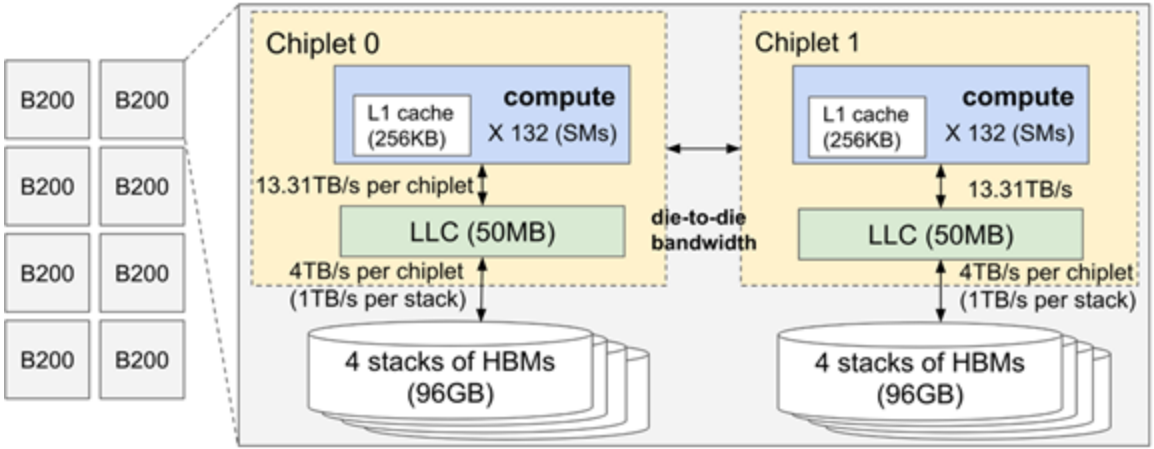


Multi-Chip-Module (MCM) Architecture

NVIDIA B200 has two chiplets

AMD MI300A APU has a mix of three CPU and six GPU chiplets

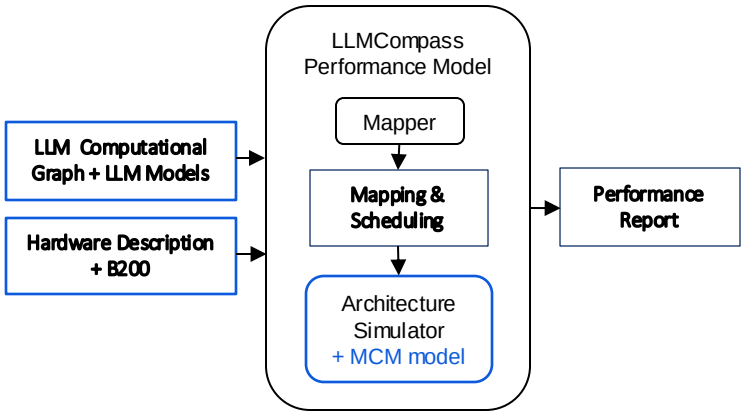
Die-to-die (D2D) bandwidth is a key performance factor on chiplet-based systems



Experimental Setup

Simulation: Enhanced LLMCompass*, HW evaluation framework

Models Investigated: LLaMA3-70B, GPT3-175B, LLaMA3-400B



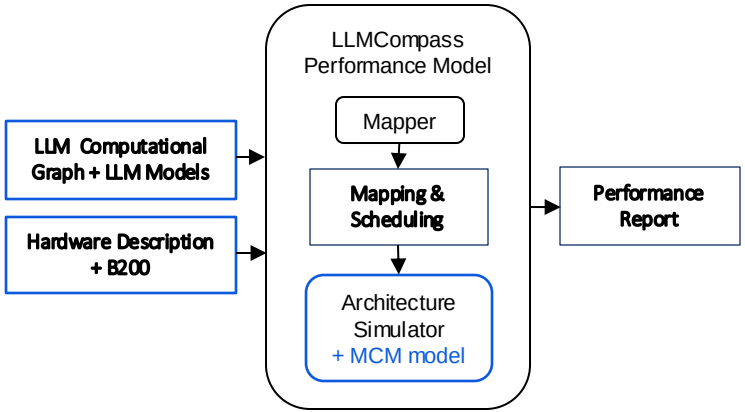
*[H. Zhang et al. ISCA'24]

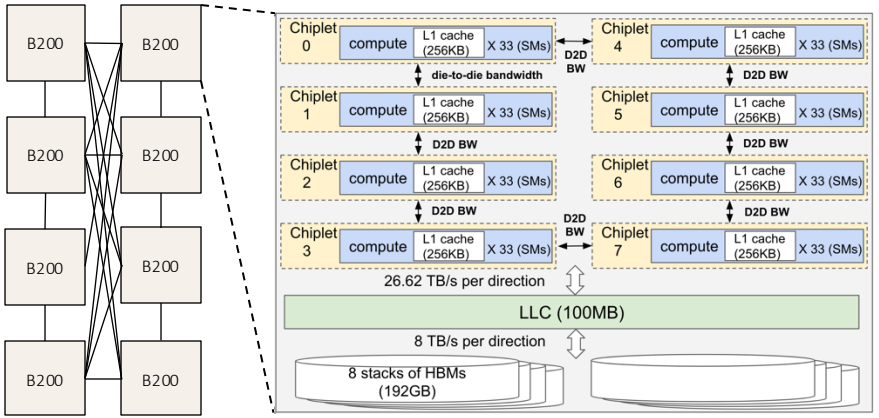
Experimental Setup

Simulation: Enhanced LLMCompass*, HW evaluation framework

Models Investigated: LLaMA3-70B, GPT3-175B, LLaMA3-400B

System Configuration: 8-device setup connected via NVLink5





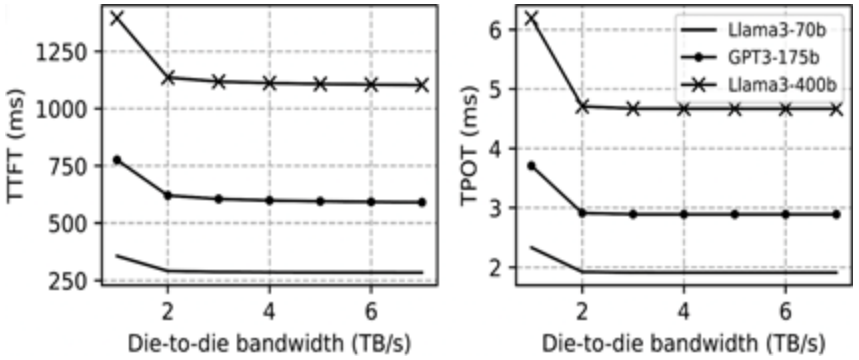
*[H. Zhang et al. ISCA'24]

D2D Impact on LLM

Low D2D BW negatively impacts latency

More sensitive to a larger model

Scales with increasing D2D BW until bidirectional D2D BW reaches the HBM BW



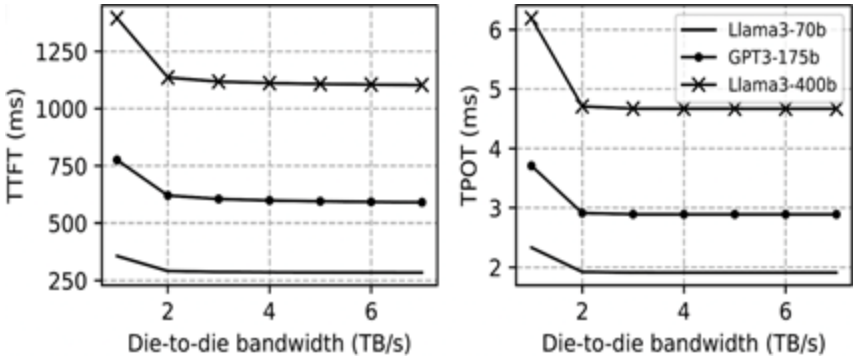
* Assumption: data is uniformly distributed across the chiplets

D2D Impact on LLM

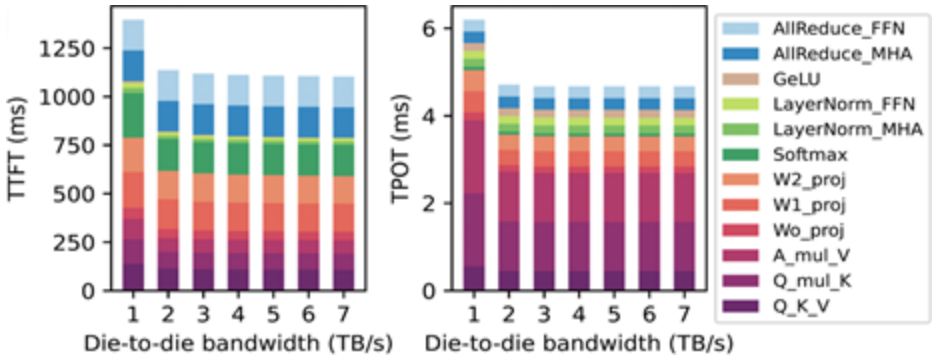
Low D2D BW negatively impacts latency

More sensitive to a larger model

Scales with increasing D2D BW until bidirectional D2D BW reaches the HBM BW



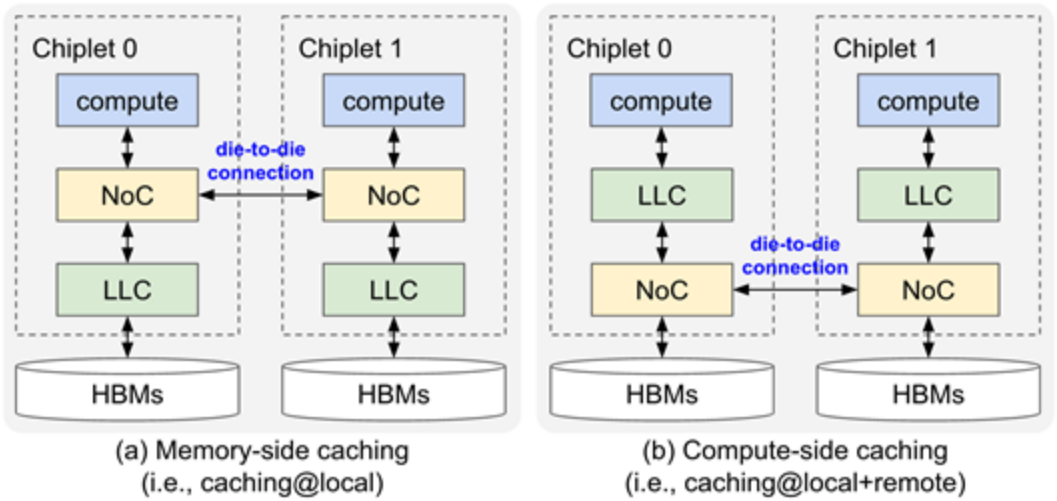
Matmul is highly sensitive to D2D (Tensor reads/writes from LLC/HBM)



* Assumption: data is uniformly distributed across the chiplets

Caching Policies on MCM

LLC only caches data from HBM attached to the same chiplet



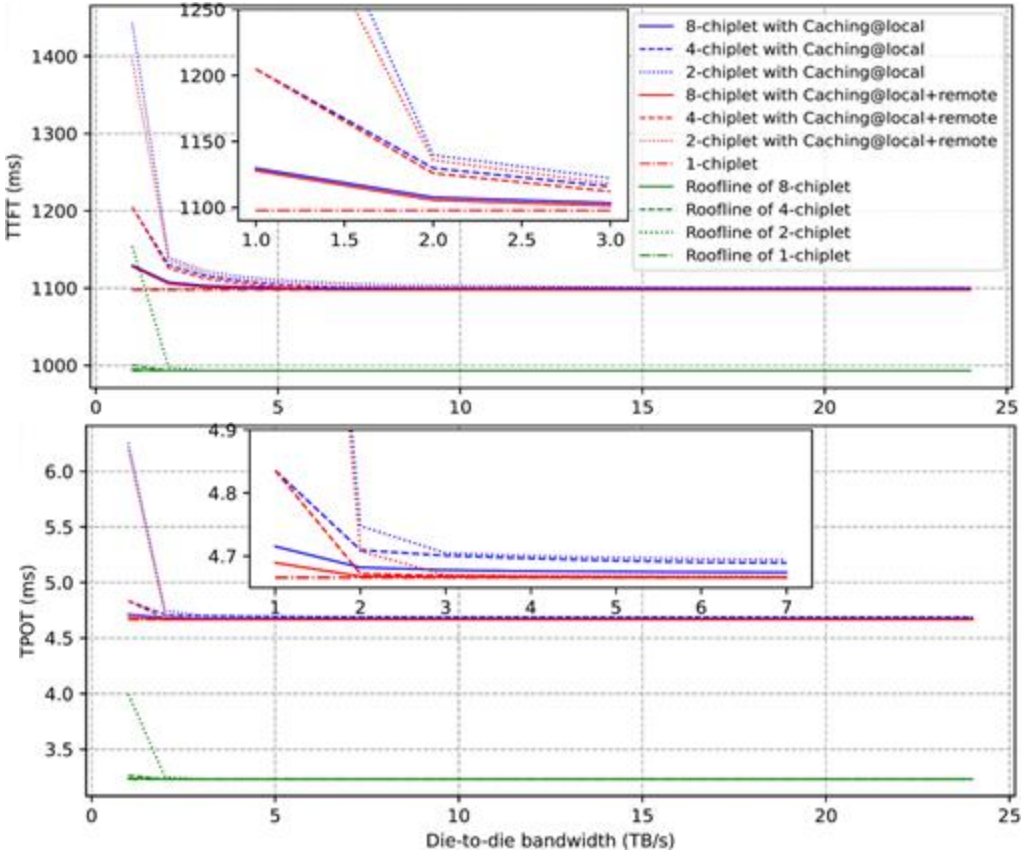
LLC caches data from HBM attached to other chiplets

MCMs x Caching Policies

Compute-side caching reduces D2D traffics

More chiplets within a single device
→ augmented cumulative D2D BW

HBM BW becomes a bottleneck after the cumulative D2D BW reaches the HBM BW



Takeaways

1. Broader D2D BW facilitates the transfer of substantial data volumes between chiplets

2. More chiplets within a single device → augmented cumulative D2D BW

3. Compute-side caching increases data reuse on local, reducing D2D traffics