Memory-Stacked GPU Architecture for LLM Inference Serving

Anonymous Authors

1. Abstract

With rapid advances in semiconductor packaging, heterogeneous 3D stacking is approaching commercial viability. GPUs present an attractive substrate for 3D packaging design exploration, for their relatively simple architectures and more focused applications. This work examines the implications of memory stacked GPU designs for LLM inference serving. Specifically, we analyze how the stacked memory can effect the memory bandwidth, capacity and latency characteristics of the system, and how they in turn impact inference token generation.

2. 3D Stacking and GPU

Heterogeneous 3D stacking sits on the verge of commercial maturity. Its predecessors, homogeneous 3D stacking and 2.5D stacking, have seen extensive commercial use, in products such as HBM and chiplet architecture [1], [2]. Major semiconductor manufacturers have announced their commercial 3D fabrication roadmaps [3], [4]. The rapid advancement of new process technology motivates architectural changes.

Redesigning a memory-stacked GPU serves as a good starting point for our design exploration. Since memory and logic are clearly seperated, a 3D design requires little changes. Moreover, the memory layer design can reference existing HBM layouts, further lower the design effort. GPUs contain large numbers of duplicated logic units, each one much simpler than CPU cores. As accelerators, they have more focused use cases, most dominant of which nowadays is LLM inference.

3. Memory Stacking for LLM Inference

LLM inference has two phases: prefill (prompt processing) and decode (token generation). The dominant phase, decode, is memory bound [5]. Therefore, a memory-stacked design brings significant benefit.

Memory stacking affects the memory system characteristics in three ways: increased bandwidth, increased capacity and reduced latency. We examine how each factor affects LLM inference.

3.1. Memory Bandwidth

As described earlier, the dominant operations during inference are bounded by memory bandwidth. In other words, increasing the memory bandwidth of the device can decrease the execution times of each kernel, which in turn decreases the execution time of inference. Thus token generation becomes faster as a direct function of device memory bandwidth.

3D stacked memory can deliver significantly higher bandwidth than HBM [6]. Adopting a stacked memory design benefits LLM inference greatly. On the first order, the execution time of each layer is directly proportional to the memory bandwith. Doubling the memory bandwith yields a 2x faster inference pipeline.

3.2. Memory Capacity

LLM inference workloads seek to maximize token throughput. At fixed memory bandwidth, the execution of each matmul and attention kernel is fixed. Therefore, the turn-around time of each prompt cannot be shortened. To circumvent this physical limit, inference serving frameworks rely on batch processing multiple prompts to increase the aggregated throughput.

Under batching, the execution times of each matmul kernels roughly stay constant, but the execution times of attention kernels increase as a function of the total sequence length. At the cost of longer per-user latency, batching improves the overall throughput by generating multiple tokens at each time step. On average, the compute time per token reduces.

However, per user serving time cannot grow infinitely. To ensure the user experience, inference services place upper limits to the per user token generation time. Benchmark tests such as MLPerf limits the per user serving time lower bound to ensure user experience [7].

Since the execution time is dictated by the memory bandwidth and the total token sequence length, for a fixed memory bandwith, there is an upper limit to the total token sequence length, and therefore an upper limit to the usful memory capacity given the available bandwith.

3.3. Memory Latency

GPUs are designed to hide the latency of memory access through the use of threading. As a result, the latency reduction from stacking provide limited benefit for large, well-optimized kernels such as matmul and attention. Our experiments corroborate with this intuition.

References

[1] N. Corporation. (2026) Nvidia h100 tensor core gpu. Accessed 2026-03-14. [Online]. Available: https://www.nvidia.com/en-us/data-center/h100/

[2] I. Corporation. (2025, October) Intel unveils panther lake architecture: First ai pc platform built on 18a. Accessed 2026-03-14. [Online]. Available: https://newsroom.intel.com/client-computing/intel-unveils-panther-lake-architecture-first-ai-pc-platform-built-on-18a

[3] TSMC. (2026) Tsmc 3dfabric: 3d silicon stacking and advanced packaging technologies. Accessed 2026-03-14. [Online]. Available: https://3dfabric.tsmc.com/english/dedicatedFoundry/technology/3DFabric.htm

[4] I. Corporation, "Foveros direct 3d: Technology brief," Tech. Rep., November 2025, accessed 2026-03-14. [Online]. Available: https://www.intel.com/content/dam/www/central-libraries/us/en/documents/2025-11/foveros-direct-3d-tech-brief.pdf

[5] H. Atmer, Y. Yao, T. Voigt, and S. Kaxiras, "Prefill vs. decode bottlenecks: Sram-frequency tradeoffs and the memory-bandwidth ceiling," *arXiv*, 2025, accessed 2026-03-14. [Online]. Available: https://arxiv.org/abs/2512.22066v1

[6] Unknown, "Cf 2018 proceedings," Federal University of Paraná (UFPR), Tech. Rep., 2018, accessed 2026-03-14. [Online]. Available: https://web.inf.ufpr.br/mazalves/wp-content/uploads/sites/13/2019/10/cf2018.pdf

[7] MLCommons, "Mlperf inference rules: Benchmarks section," https://github.com/mlcommons/inference_policies/blob/8adac316/inference_rules.adoc#41-benchmarks, 2024, accessed: 2026-03-14.