Ninth Annual Conference on Machine Learning and Systems

TokenWeave: Efficient Compute-Communication Overlap for Distributed LLM Inference

Raja Gond, Nipun Kwatra, and Ramachandran Ramjee
Microsoft Research India

MLSys 2026

Motivation: Communication Overhead in Tensor-Parallel LLM Serving

Tensor Parallelism (TP) is the standard approach for serving modern LLMs across multiple GPUs within a single node. Each attention and FFN sublayer requires an AllReduce that lies on the critical path of the forward pass.
Despite high-speed NVLink interconnects:

- AllReduce costs 9–23% of end-to-end latency
- RMSNorm (after AllReduce) adds another 4–9%

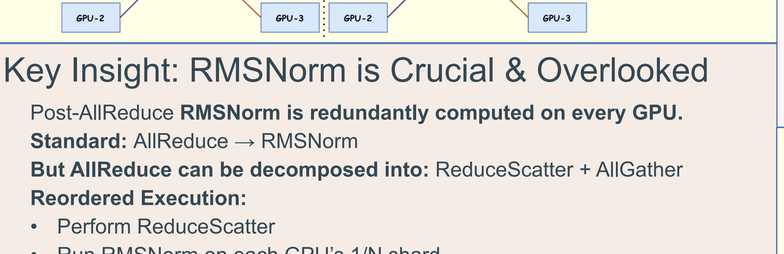
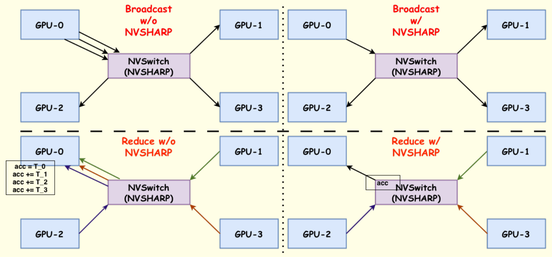


Figure: AllReduce & RMSNorm overheads on 8×H100 DGX (vLLM v0.8.5).

The Fundamental Challenge

There are two main challenges:
1) Overlap techniques hide communication by decomposing computation into subtasks; however, this decomposition itself introduces overhead.
Solution: Use a two-way GPU-wave-aware approach to create a pipeline.
2) Communication itself requires many GPU streaming multiprocessors (SMs), which would otherwise be available for computation.
Solution: Leverage NVSHARP/Multimem feature available on modern NVIDIA GPUs, such as Hopper and Blackwell.



Key Insight: RMSNorm is Crucial & Overlooked

Post-AllReduce **RMSNorm is redundantly computed on every GPU.**
Standard: AllReduce → RMSNorm
But AllReduce can be decomposed into: ReduceScatter + AllGather
Reordered Execution:

- Perform ReduceScatter
- Run RMSNorm on each GPU's 1/N shard
- Use AllGather to distribute the post-norm values to all GPUs

Benefit: Reduces RMSNorm work by **N×**
Challenge: Naïve decomposition doesn't work because splitting AllReduce into ReduceScatter + AllGather is inefficient (i.e. RS + AG > AR).
Solution: A custom fused AllReduce–RMSNorm kernel that eliminates the extra memory traffic and leverages NVSHARP capabilities!

TokenWeave: System Overview

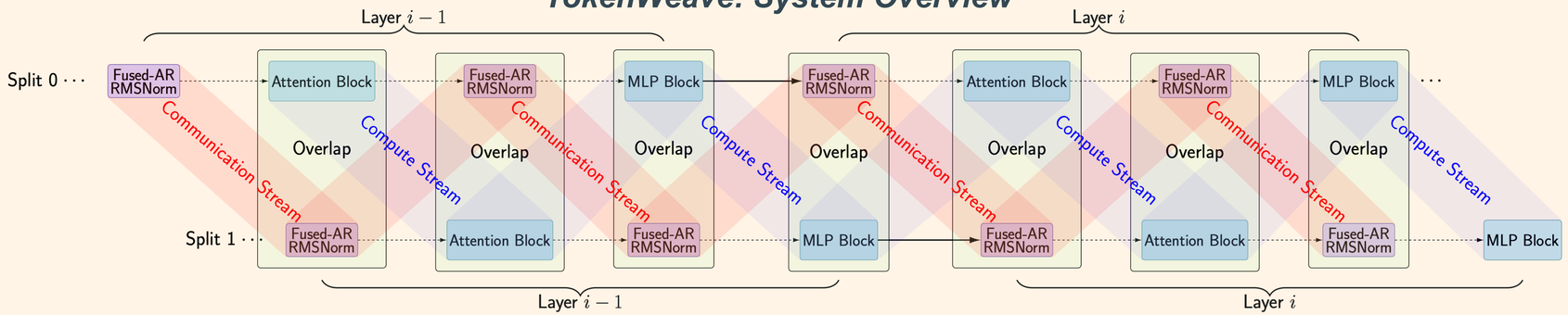
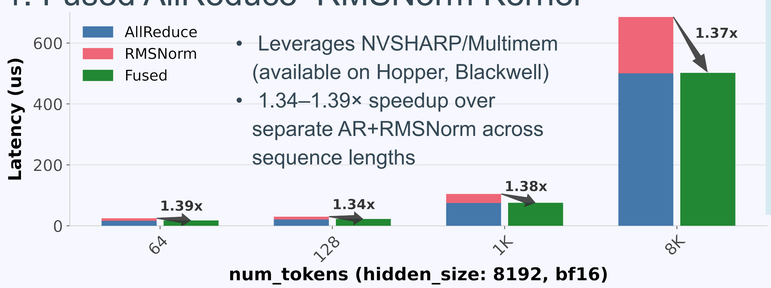


Figure: **TokenWeave** partitions the input batch into two token splits. Fused AllReduce–RMSNorm of one split overlaps with computation of the other via separate CUDA streams.

1	2	3
Fused AllReduce–RMSNorm Kernel Minimises HBM traffic; uses only 2–8 SMs	Wave-Aware Token Splitting Avoids wave-quantization overhead from splitting	Adaptive Overlap Enabling Full overlap ≥ threshold; fused kernel only below

1. Fused AllReduce–RMSNorm Kernel

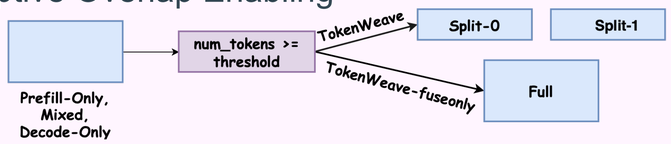


- Leverages NVSHARP/Multimem (available on Hopper, Blackwell)
- 1.34–1.39× speedup over separate AR+RMSNorm across sequence lengths

AllReduce + RMSNorm = ReduceScatter + RMSNorm // G + AllGather
(G — Number of GPUs)

The fused kernel carefully fuses ReduceScatter, RMSNorm, and AllGather by minimizing HBM traffic. Further, it also helps achieve efficient overlap while reserving only 2–8 SMs by leveraging NVSHARP .

3. Adaptive Overlap Enabling



Full TokenWeave with splitting and overlap is enabled only for higher values of num_tokens. For smaller num_tokens, where splitting can result in higher overheads, we only enable the fused kernel.

Evaluation: Up to 1.19× Throughput Improvement

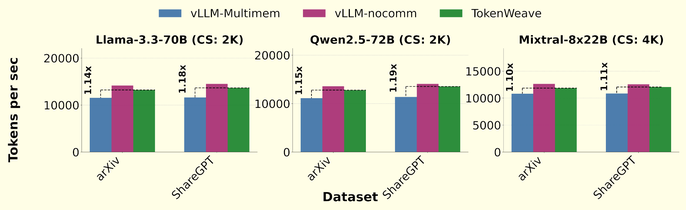


Figure: End-to-end throughput on ShareGPT & ArXiv traces across three models on 8×H100.

As shown, in the 8×H100 setting, TokenWeave continues to show substantial throughput gains, effectively recovering most of the communication overhead.

2. Wave-Aware Token Splitting

Splitting at wave boundaries eliminates quantization overhead.

Choose split sizes so combined GPU waves = waves of the unsplit batch.

Evaluation: Up to 1.28× Latency Speedup

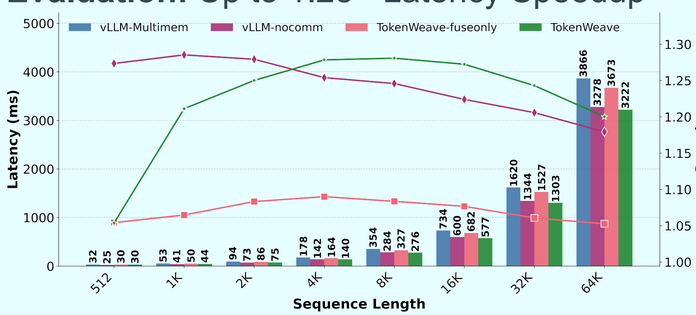


Figure: Inference latency of Llama-3.3-70B on an 8×H100 DGX system for various sequence lengths. vLLM-Multimem corresponds to vLLM with an optimized AllReduce implementation using Multimem and NVSHARP support. vLLM-nocomm is a counterfactual baseline corresponding to only the computation time without any communication.

TokenWeave achieves up to 1.28× speedup. Even at shorter sequence lengths, TokenWeave provides significant gains, e.g., 1.2× at a sequence length of 1K tokens. At sequence lengths ≥ 4K, TokenWeave outperforms vLLM-nocomm by not only recovering all communication overhead but also providing additional gains due to its AllReduce–RMSNorm fused kernel.