



THE UNIVERSITY  
*of* EDINBURGH



# WaferLLM: Large Language Model Inference at **Wafer Scale**

**Congjie He**<sup>1</sup>, Yeqi Huang<sup>1</sup>, Pei Mu<sup>1</sup>, Ziming Miao<sup>2</sup>, Jilong Xue<sup>2</sup>, Lingxiao Ma<sup>2</sup>, Fan Yang<sup>2</sup>, Luo Mai<sup>1</sup>

University of Edinburgh<sup>1</sup>, Microsoft Research<sup>2</sup>



# The challenge for scaling LLM inference

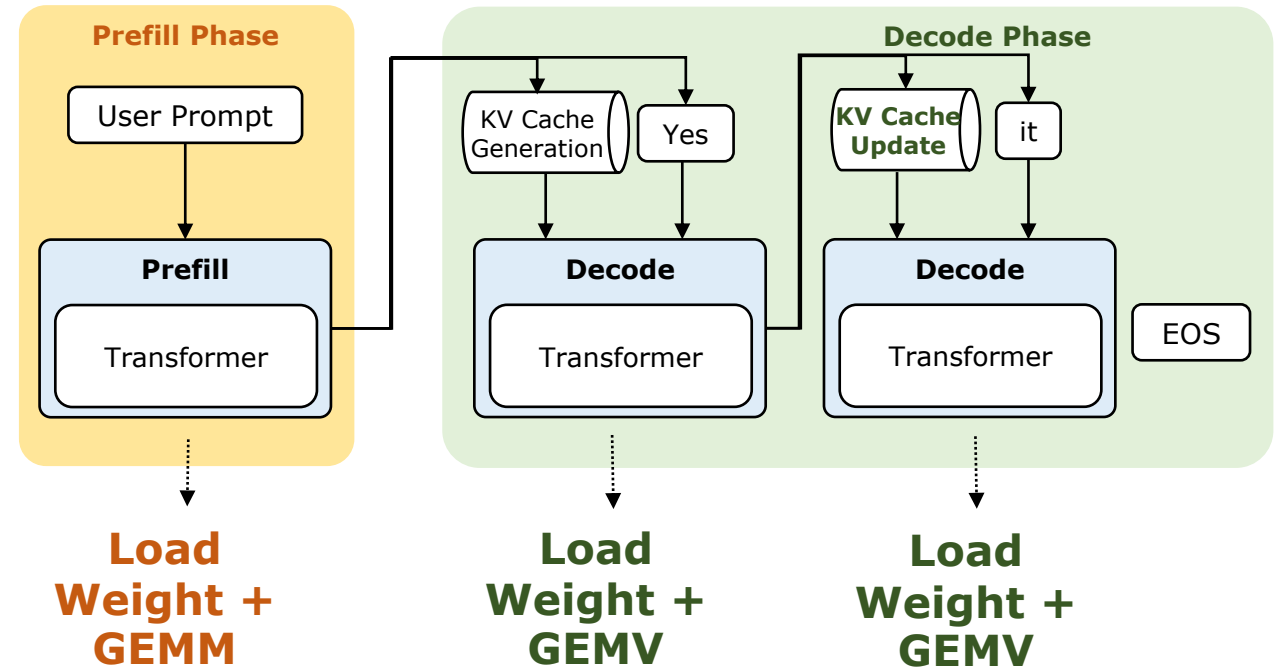
LLM needs tremendous bandwidth

- Massive model weights & KV cache are **repeatedly accessed**
- Example: ~100s TB/s for DeepSeek-R1 (10K tokens/s per request)

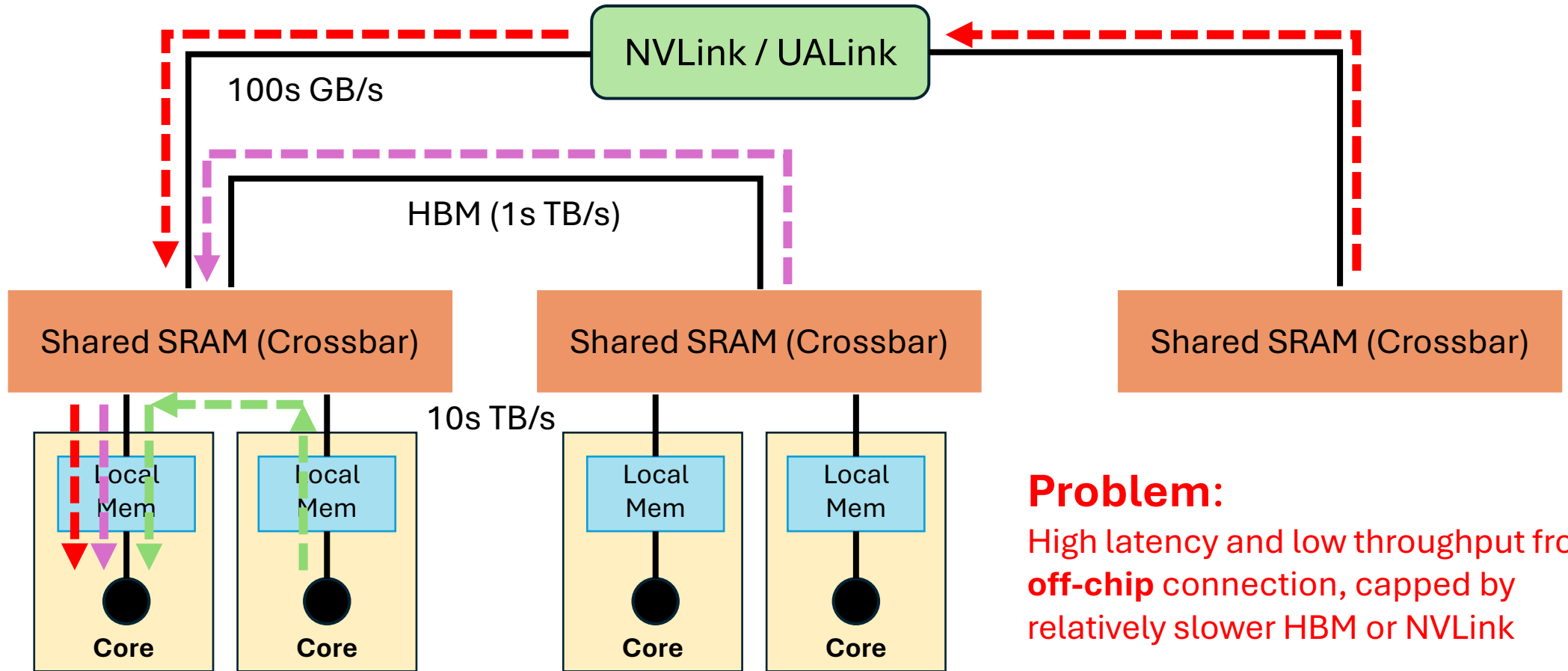
**Test-time Scaling** and **AI Agents** need more

- Deep thinking for each request
- Agent-agent interaction
  - Millions of tokens/s per request

LLM inference process



# Today's AI systems primarily use **off-chip scaling**



# Efficient **on-chip scaling** with wafer-scale integration

## Massive cores in a mesh-like topology

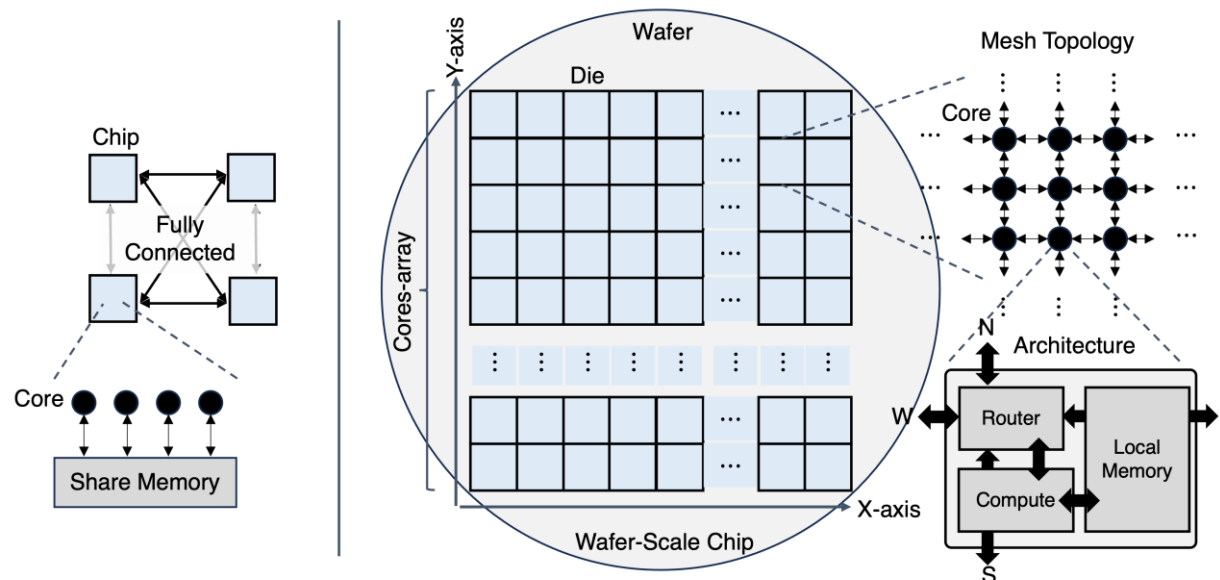
- 10s of times bigger chip size than a typical die

## Efficient integration on a single wafer

- Memory (**~10s GB SRAM**)
- Compute (**~10s PFLOPS dense FP16**)
- Memory bandwidth (**~10s PB/s**)
- NoC bandwidth (**~100s Pbits/s**)

## Complementary to off-chip scaling

- **Cerebras**: Memory-X clusters (**PB-scale**) [1]
- **Tesla Dojo**: HBM/DRAM on switch [2]



Off-chip scaling via  
NVLink/IB

On-chip scaling via wafer-  
scale integration

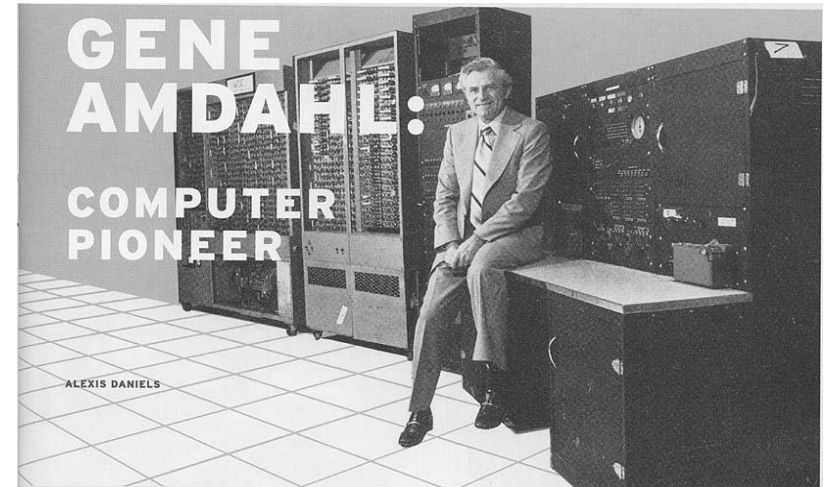
*Imagine to fit the entire rack into a wafer*

[1] <https://www.cerebras.ai/press-release/cerebras-systems-announces-worlds-first-brain-scale-artificial-intelligence-solution>

[2] [https://hc2024.hotchips.org/assets/program/conference/day2/17\\_HC2024\\_Tesla\\_TTPoE\\_v5.pdf](https://hc2024.hotchips.org/assets/program/conference/day2/17_HC2024_Tesla_TTPoE_v5.pdf)

# Wafer-scale Integration. Why not before?

- Gene Amdahl shared a similar observation [1]
  - The pioneer of mainframe machines
  - The author of Amdahl's Law
- Amdahl co-founded Trilogy Systems
  - Attempted to design the first wafer-scale chips
  - The biggest investment (\$200M) in Silicon Valley in the 1980s
- Trilogy Systems failed due to
  - **Low yields at wafer-scale**
  - **Weak market demand**

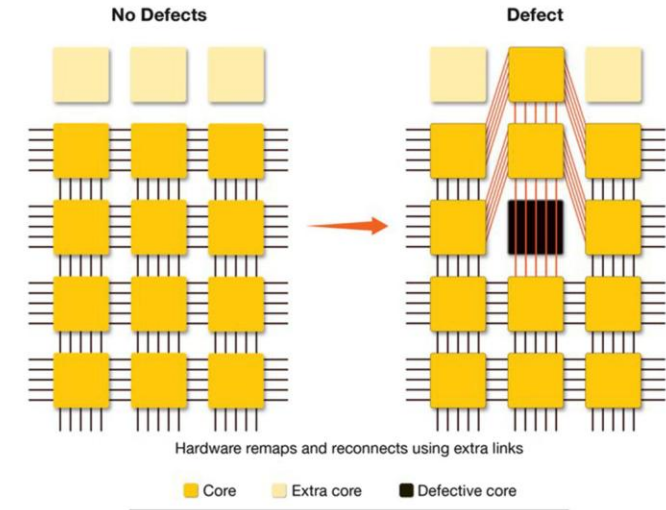


[1] <https://spectrum.ieee.org/whats-better-than-40-gpubased-servers-a-server-with-40-gpus>

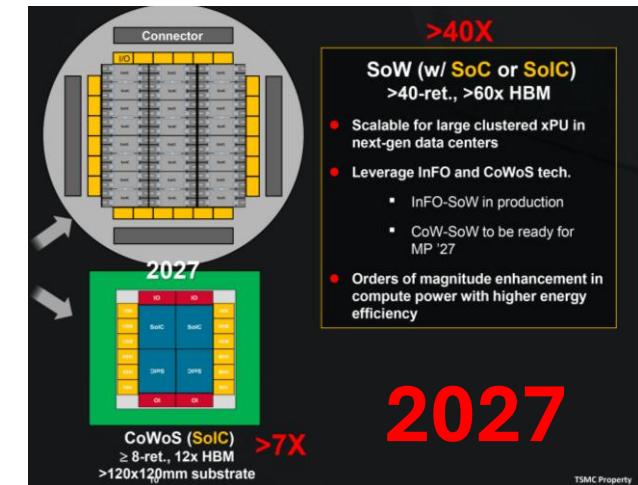


# What has changed in 40 years?

- AI chases extreme efficiency/performance
- Hardware remapping
  - Cores are made small to limit chip defect scope
  - Bypass the defective core with redundant wires
- Remapping becomes viable and cost-effective
  - **Small yet fast cores:** WSE-3 core = **0.7%** size of NV H100 SM
  - **High yield: 93%** core active (WSE-3) vs **92%** (NV H100)
- A wave of wafer-scale computers is coming
  - **>40X** compute and bandwidth expected **by 2027** [2]
  - Advanced packaging (CoWoS), 3DIC (TSMC SoIC)



Example of hardware remapping [1]



TSMC Roadmap – System-on-Wafer[2]

[1] <https://www.cerebras.ai/blog/100x-defect-tolerance-how-cerebras-solved-the-yield-problem>

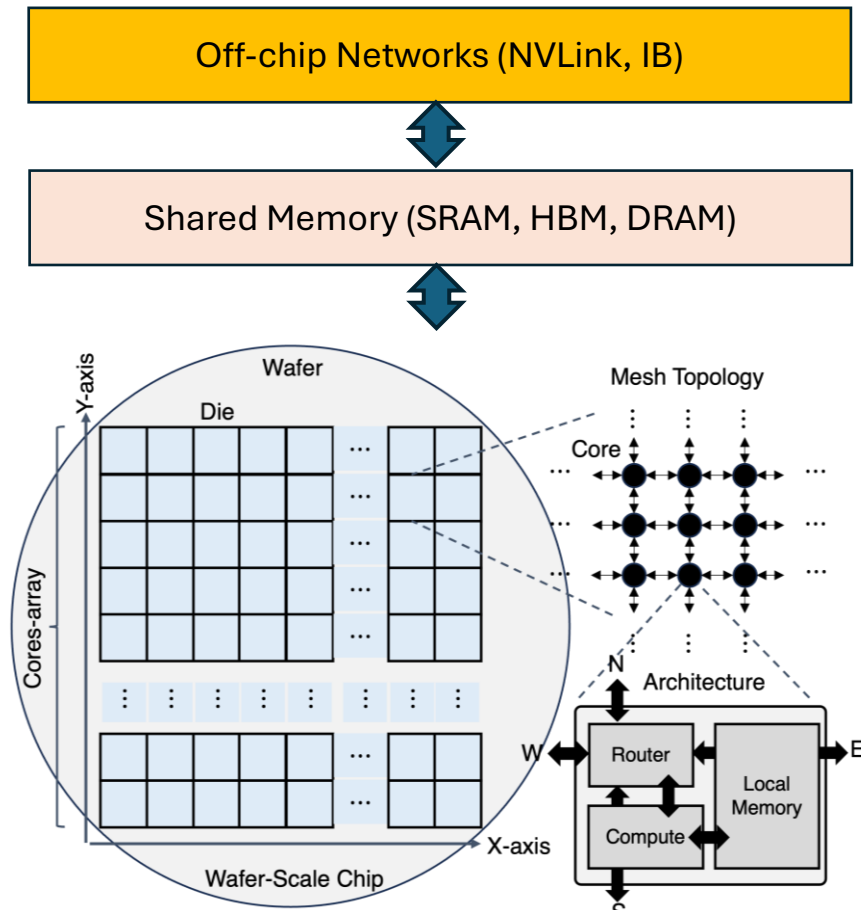
[2] <https://www.tomshardware.com/tech-industry/tsmc-to-go-3d-with-wafer-sized-processors-cow-sow-system-on-wafer-technology-allows-3d-stacking-for-the-worlds-largest-chips>

# Today's wafer-scale integration has shown great promise

	System-on-Die	System-on-Wafer	
Area	Typically 858 mm <sup>2</sup>	Typically 73062 mm <sup>2</sup>	
#Transistors (TSMC n3)	1 trillion	Up to 91 trillion	
Interconnect	PCB/RDL/SUB/WoW	Wafer	
Die-to-die efficiency	<b>~10s pJ/bit</b>	<b>~0.1s pJ/bit</b>	<b>~100x</b>
Die-to-die bandwidth	~ 1-10s TB/s	~ 10 - 100s TB/s	<b>~100x</b>
Memory Bandwidth	<b>10s TB/s</b> (crossbar)	<b>10s PB/s</b> (aggregated via mesh)	<b>~1,000x</b>
Off-chip memory	10s - 100s GB HBM	10s TB DRAM via Ethernet <b>10s TB HBM/DRAM via TSMC SoW in 2027</b>	

- **Emerging wafer-scale systems:** Cerebras, Tesla Dojo, NVIDIA and more reported by TSMC
- **Growing adoption:** Perplexity, Mixtral, Meta AI, G42, ...

# Are LLM systems ready for wafer-scale chips?



## Extensive research on scaling LLM with off-chip networks

- **Topology:** Clos, 3D-Torus
- **System:** Megatron-LM
- **Multi-dimensional Parallelism:** TP, PP, DP, EP
- **Communication Operator:** Ring allreduce, All-to-All for MoE

## Extensive research on LLM with on-chip shared memory

- **Operator:** FlashAttention, MLA, PageAttention,
- **Compiler:** Ladder [OSDI'24], and T10 [SOSP'24]

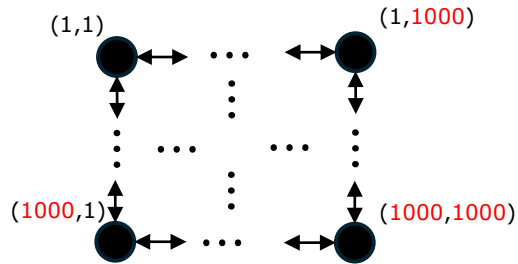
## Wafer-scale AI software remains largely unexplored

- Existing NoC research targets CPUs and small scale (up to 100s)
- Suffer severe communication bottlenecks
- ...



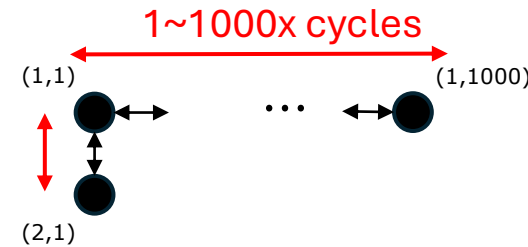
# PLMR – a hardware model to describe wafer-scale chip

## 1. Million-scale Parallelism (PLMR)



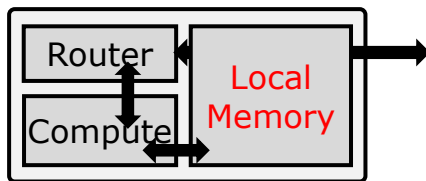
From hundreds of parallelism in a crossbar to millions of parallelism in a mesh

## 2. Highly non-uniform access Latency (PLMR)



From shared memory and small NUMA to large-scale non-uniform memory

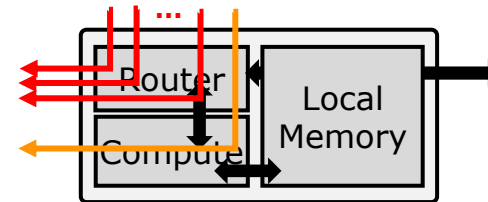
## 3. Constrained local Memory (PLMR)



100s KB – 1s MB

From coarse-grained tile pipeline to fine-grained tile pipeline

## 4. Constrained Routing resources (PLMR)



Only support 10s routing entries

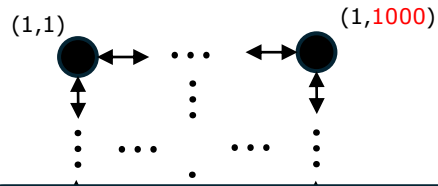
From centralized routing to decentralised NoC routing

→ Routing on NoC

→ Routing on Compute Engine

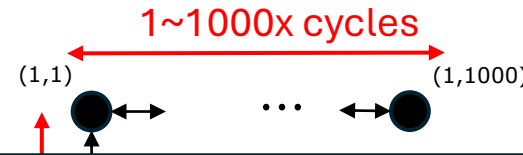
# PLMR – a hardware model to describe wafer-scale chip

## 1. Million-scale Parallelism (PLMR)



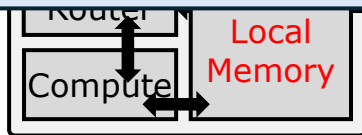
From hundreds of parallelism in a crossbar to millions of

## 2. Highly non-uniform access Latency (PLMR)



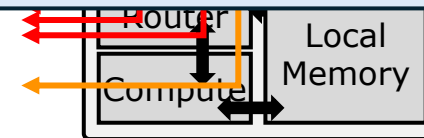
From shared memory and small NUMA to large-scale non-

**PLMR model** - The key technological shift is moving from *shared-memory architectures* to *on-chip large-scale, distributed memory systems*



100s KB – 1s MB

From coarse-grained tile pipeline to fine-grained tile pipeline



**Only support 10s routing entries**

From centralized routing to decentralised NoC routing

→ Routing on NoC

→ Routing on Compute Engine

# WaferLLM: World-first wafer-scale LLM inference system

## Goals

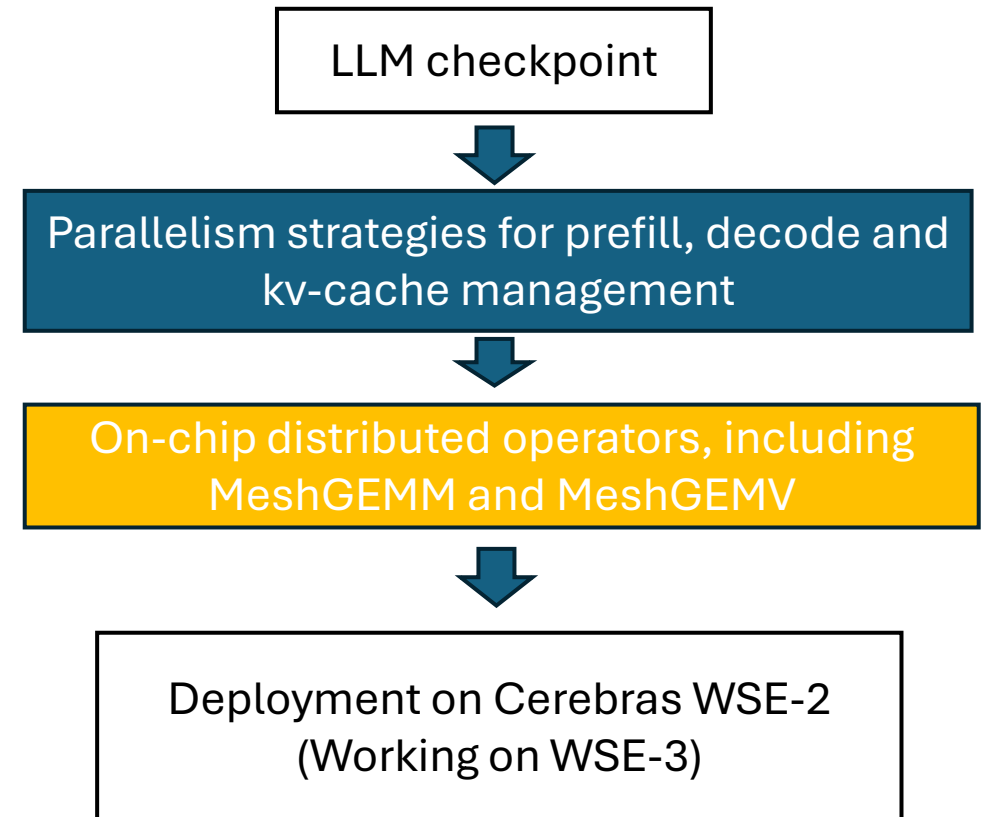
- **Design the entire stack guided by PLMR**
- Generalise across hardware backends

## Contributions

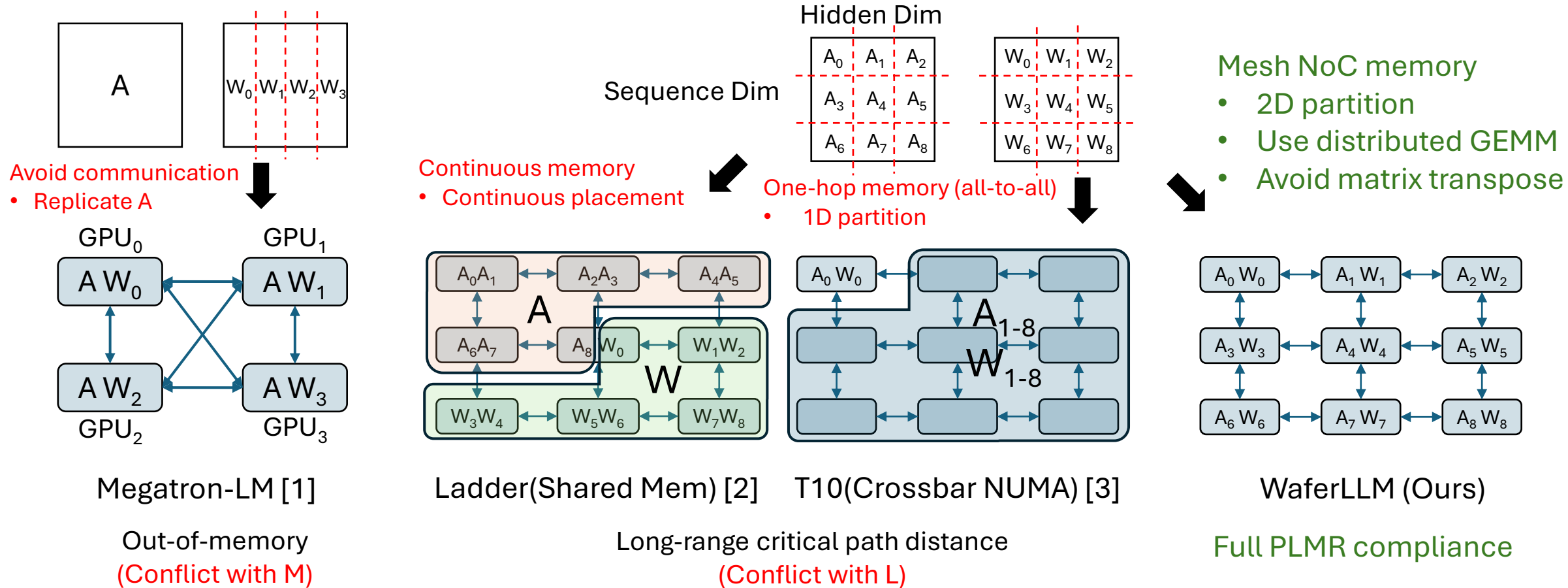
- New prefill parallelism strategies
- New decode parallelism strategies
- New KV-cache algorithm – Shift-based update
- New GEMM algorithm - MeshGEMM
- New GEMV algorithm - MeshGEMV

**First LLM inference system to reach 2700 token/s per request**

## Current WaferLLM Architecture



# How to scale **prefill** parallelism?



- Achieving **PLMR compliance** can lead to **372x** gains over Ladder with LLaMA2-13B
- Also **113x** over T10 with LLaMA2-13B

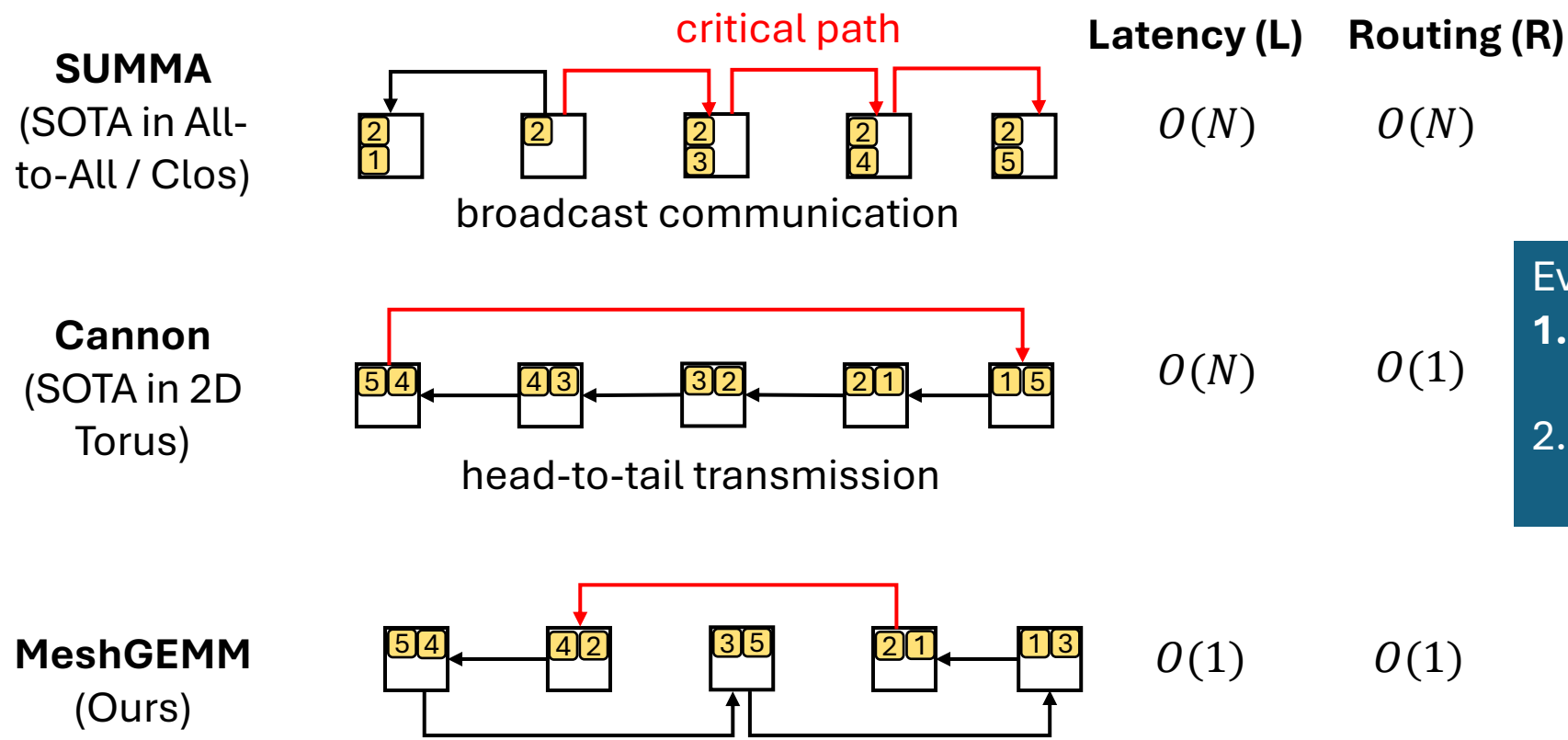
[1] Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism, <https://github.com/NVIDIA/Megatron-LM>

[2] Ladder: Enabling Efficient Low-Precision Deep Learning Computing through Hardware-aware Tensor Transformation, OSDI 2024

[3] Scaling Deep Learning Computation over the Inter-Core Connected Intelligence Processor with T10, SOSP 2024

# Accelerating prefill with MeshGEMM

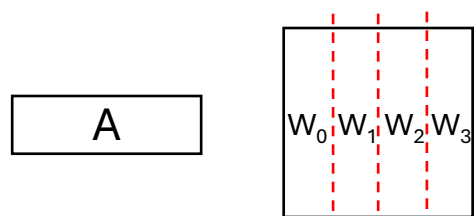
Prefill is bottlenecked by GEMM, which requires each submatrix to traverse all row (or column) cores, constrained by properties L and R.



Evaluation results:

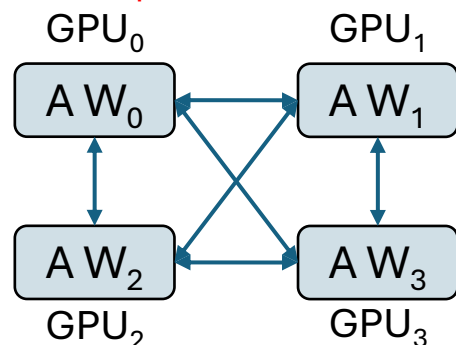
1. **1.3–2×** faster GEMM with matrix sizes from 2K to 8K
2. Reduces communication overhead by **2–5×**.

# How to scale decode parallelism?



Avoid communication

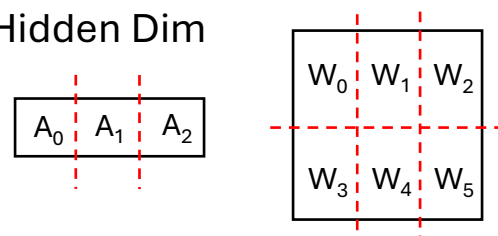
- Replicate A



Megatron-LM

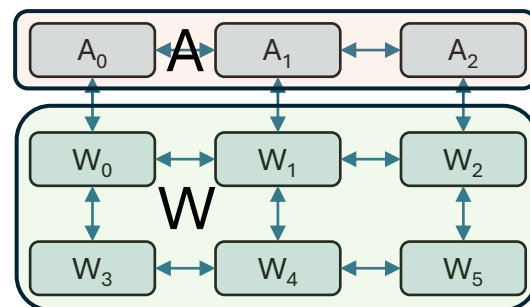
Insufficient parallelism  
(Conflict with P)

Hidden Dim



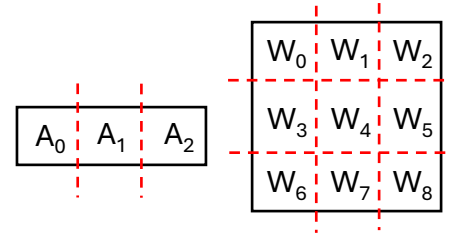
Continuous memory

- Continuous placement



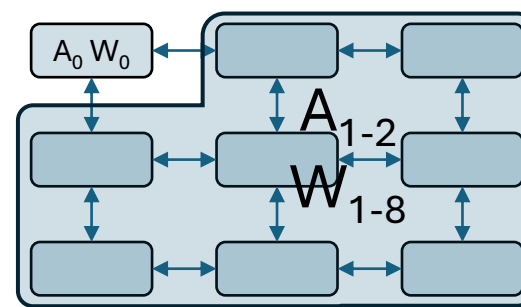
Ladder (Shared Mem)

Long-range path & insufficient parallelism  
(Conflict with L and P)

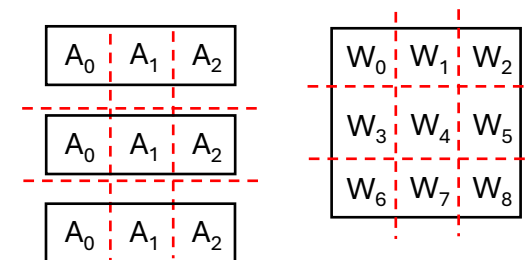


One-hop memory (all-to-all)

- 1D partition

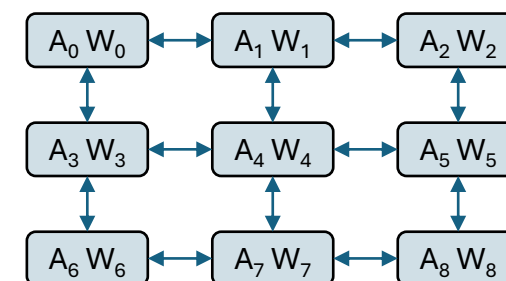


T10 (Crossbar NUMA)



Mesh NoC-based memory

- Partition & replicate
- Applied distributed GEMV
- Avoid vector transpose



WaferLLM (Ours)

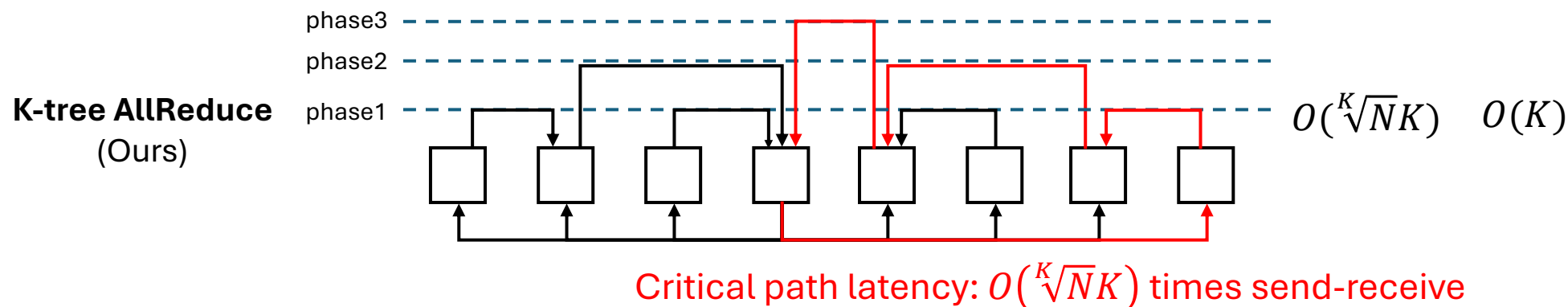
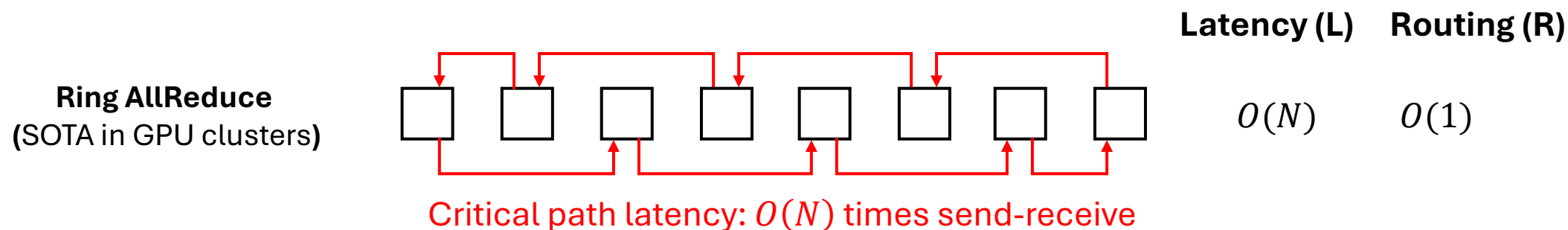
Achieving P, L, M and R

- WaferLLM outperforms Ladder by **185x** with LLaMA2-13B
- Also **6x** over T10 with LLaMA2-13B



# Accelerating decode with MeshGEMV

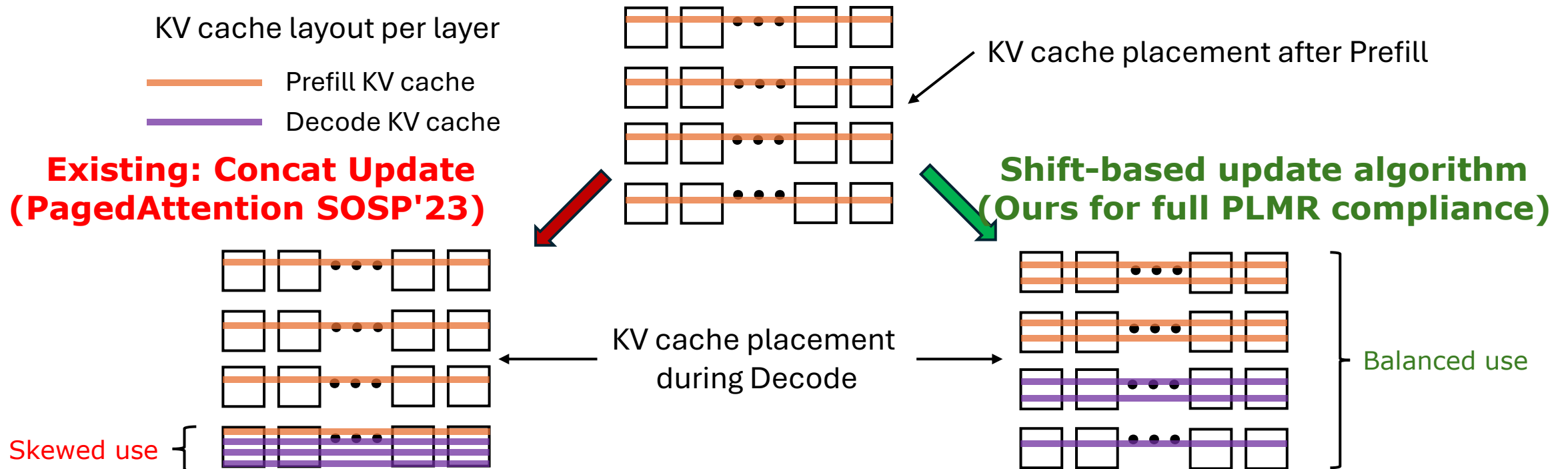
GEMV is bottlenecked by all-reduce, decided by properties L and R



**Key technique: tree-based grouped parallel reduction**

Evaluation results:  
1. **2.5-3X** faster GEMV with vector sizes from 2K to 8K.  
2. Reduces communication overhead by **1.5-6X**.

# Scale KV-cache management using **shift update**

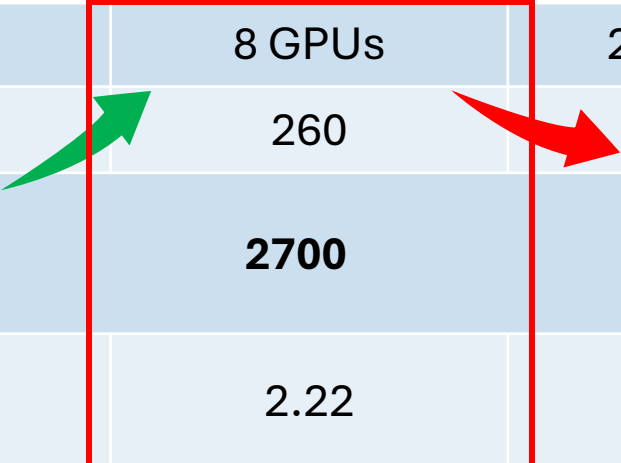


- Avoid out-of-memory caused by skewed core usage (M)
- Rebalance via data movement between adjacent cores (L)
- The maximum inference length improves **several 100x**

# Comparison with SOTA off-chip scaling on LLM inference

We compare **WaferLLM on real Cerebras WSE-2 chip (TSMC 7nm)** with **SGLang/vLLM on NVLink/IB-connected A100 GPU (TSMC 7nm)** in performance and energy efficiency

Decode (4K in, 4K out, BSZ=1)	LlaMA3-8B		
	1 GPU	8 GPUs	2x8 GPUs
SGLang (A100) Token/s per request	78	260	164
WaferLLM (WSE-2) Tokens/s per request		<b>2700</b>	
A100/WSE-2 Energy Ratio	0.92	2.22	7.02



- WaferLLM delivers **6-20x** faster than SoTA off-chip solutions on LLM model size range from 8B to 70B
- **2-2.5x** energy efficiency than GPU interconnect – currently the only one on the market beyond NVLink
- WaferLLM with wafer-scale chip **outperforms best-case off-chip scaling in both speed and efficiency**

# WaferLLM: The first step of our long journey

## **Align AI model designs with wafer-scale systems**

- New model architectures?
- New training and inference algorithms?

## **Rethink software designs**

- Operating system for wafer-scale computers?
- Distributed programming libraries?

## **Explore new hardware designs**

- Better core design?
- Better interconnect designs?

**Acknowledgement:** Hardware support from

- Edinburgh International Data Facility (EIDF)
- Edinburgh Parallel Computing Centre (EPCC)

Thank you!



<https://github.com/MeshInfra/WaferLLM>