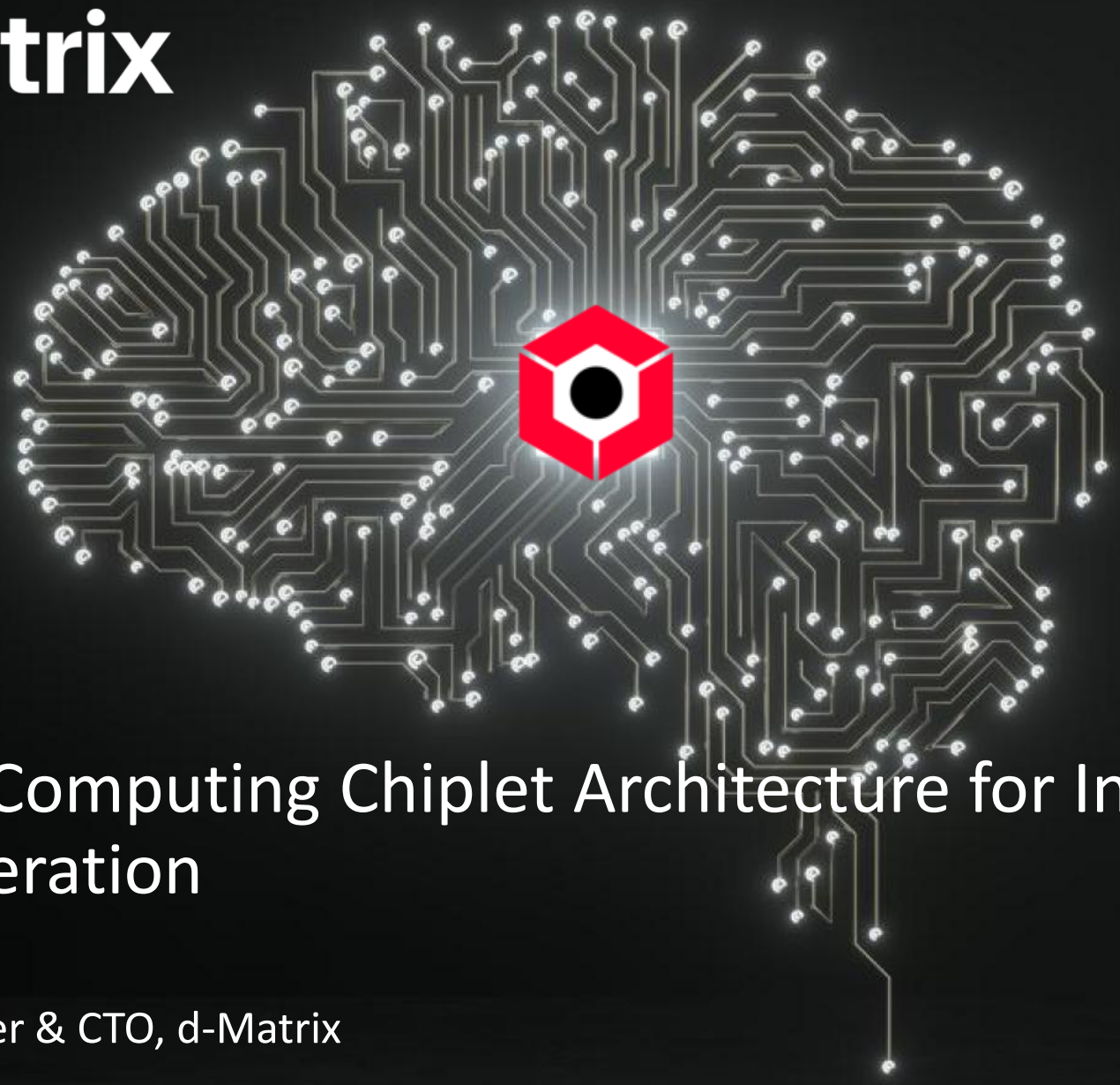
d-Matrix

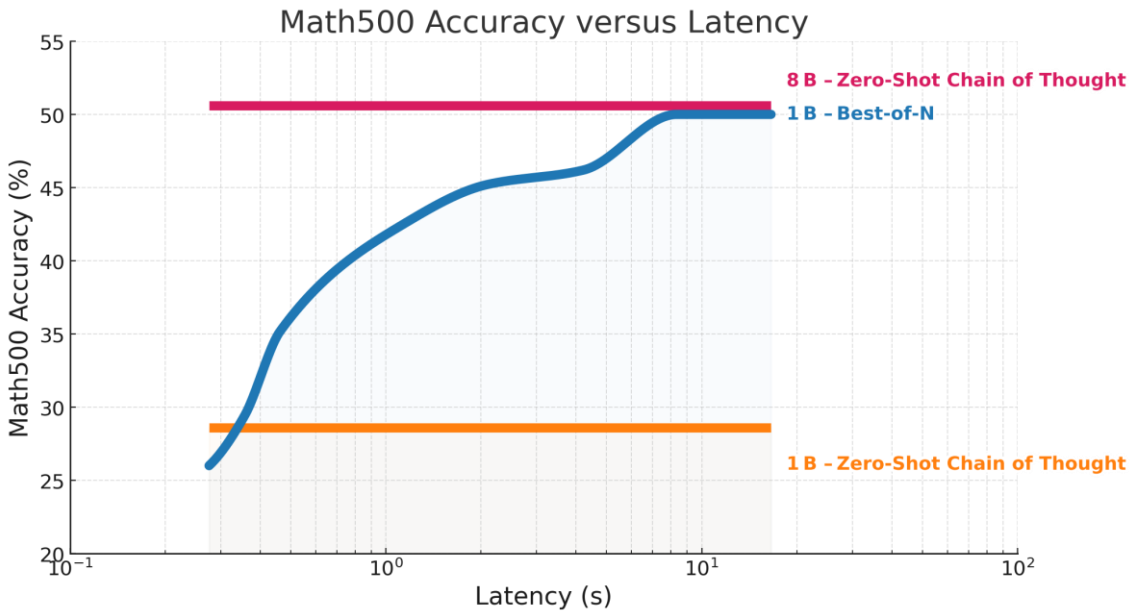


# CORSAIR:
## An In-Memory Computing Chiplet Architecture for Inference-Time Compute Acceleration

Sudeep Bhoja, Co-founder & CTO, d-Matrix
Hotchips 2025

# “Rethinking” AI Inference

- Bitter Lesson[1]: Search and Learning
- Small Reasoning Models[2,3] outperform 20x Larger LLMs
- Chain of thought 10-100x more tokens than classic LLMs
- Keeping user latency constant requires 10-100x memory BW



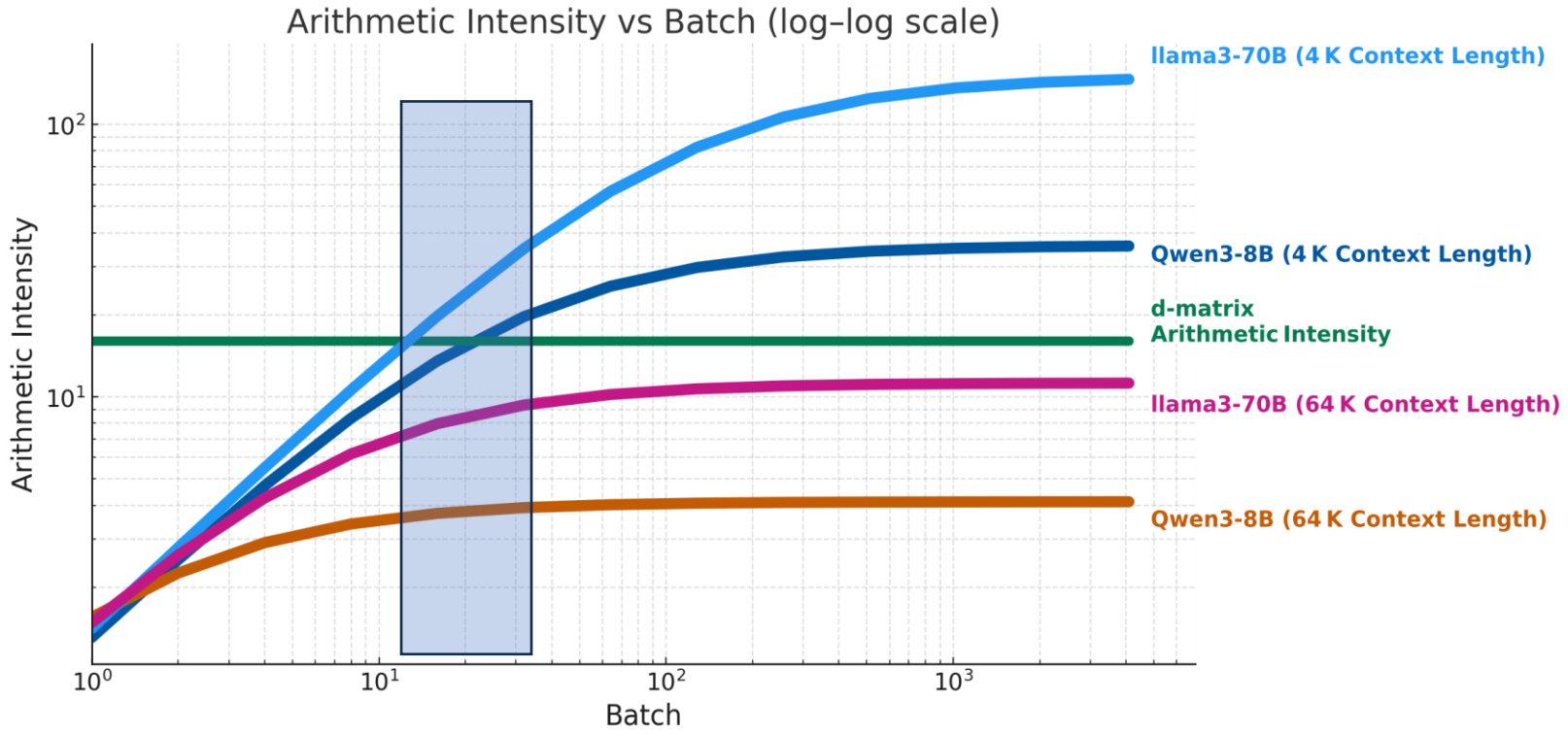

Latency measured on A100 GPU

[1]Bitter Lesson

[2]Denny Zhou, LLM-Reasoning-Stanford-CS-25.pdf

[3]Scaling test time compute

# LLM Token-generation is Memory Bound

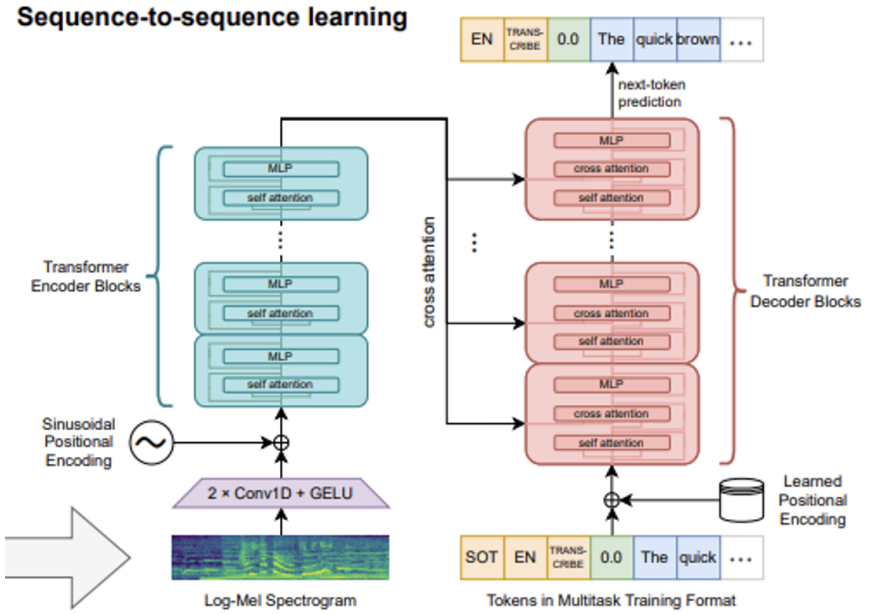



- SLO dictates small batch 16-64[1,2]
- Arithmetic intensity saturates as KV cache grows
- Token-generation is a memory bound problem!

[1]Google TPU v1

[2]CMU Latency Throughput tradeoffs

# Voice is latency-critical and even more Memory Bound

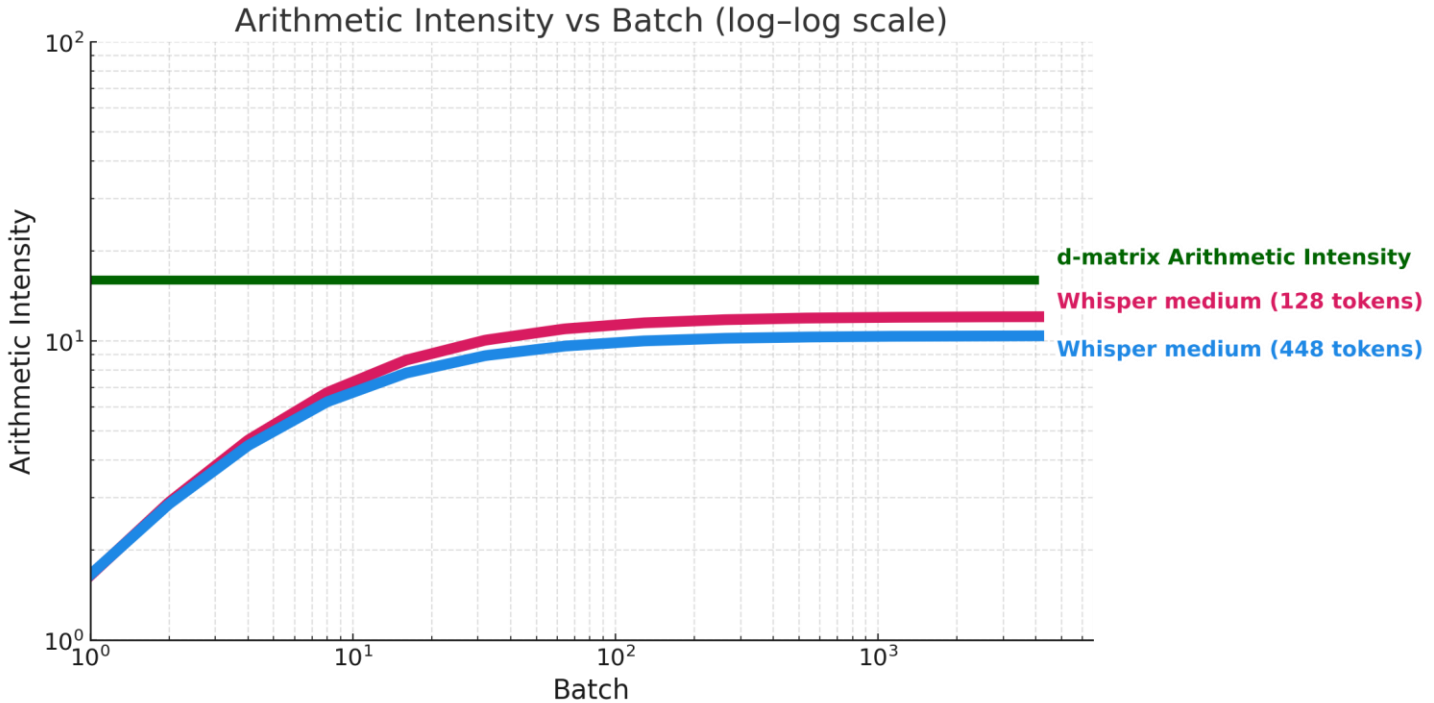



| Model | Layers | Width | Heads | Parameters |
|---|---|---|---|---|
| Tiny | 4 | 384 | 6 | 39M |
| Base | 6 | 512 | 8 | 74M |
| Small | 12 | 768 | 12 | 244M |
| Medium | 24 | 1024 | 16 | 769M |
| Large | 32 | 1280 | 20 | 1550M |

Source: OpenAI, Whisper

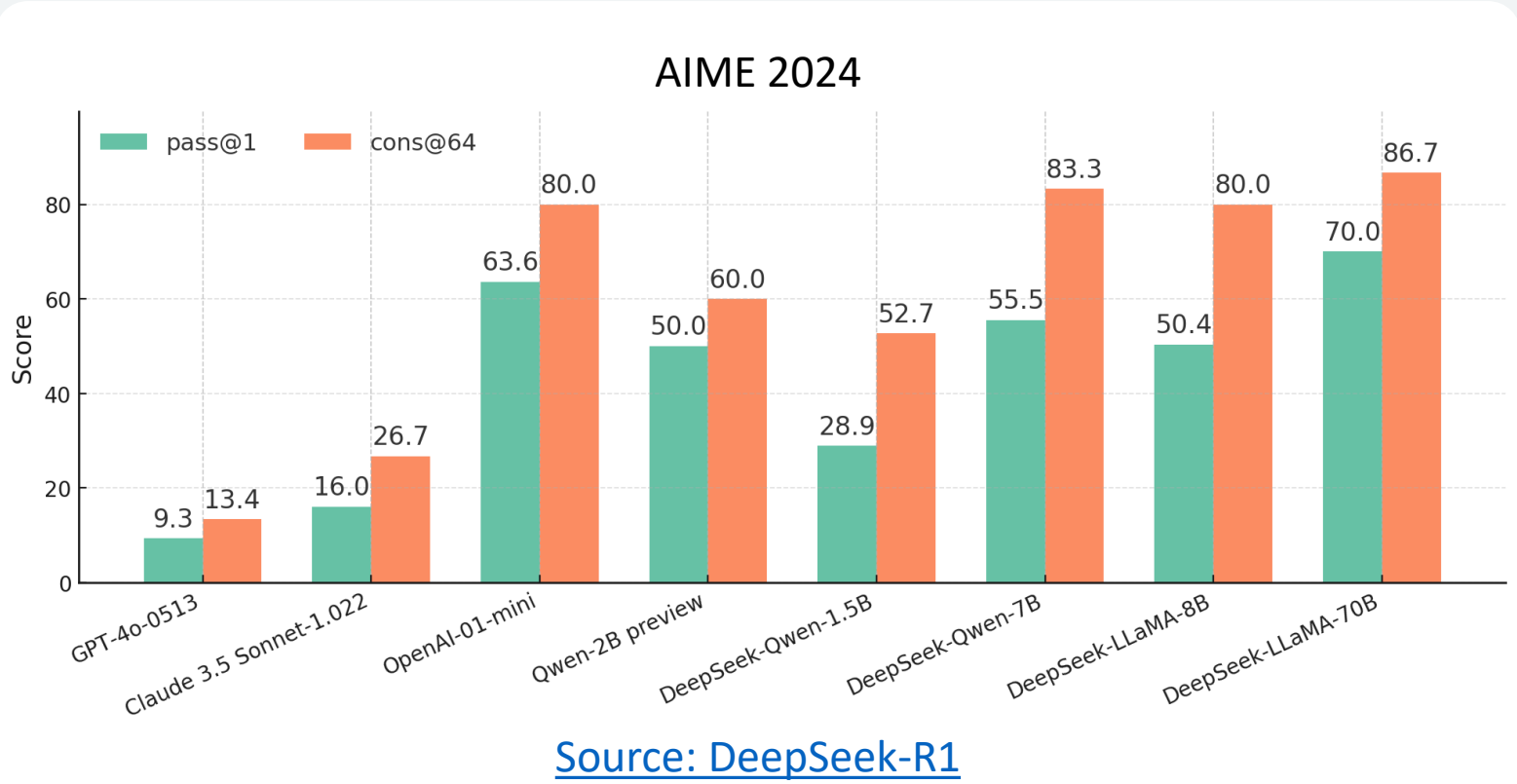
## Whisper

- Memory bound in token-generation
- Decoders include self and cross attention KV caches



# AI Agents: SLM & Rise of Inference-Time Compute

- AI agents use multiple small models to ensure task specialization
- Small models solve speech, reasoning, tool calling & code generation
  - OpenAI gpt-oss-20B
  - DeepSeek-R1-Distill
  - Qwen3 “thinking-mode” 4B
  - Google Gemma3
  - Huggingface SmolLM3
  - Whisper
- Small Language Models (SLM) → Future of Agentic AI[1]




Source: DeepSeek-R1

Small distilled models with reasoning traces *outperform* LLMs

[1] SLM Agents

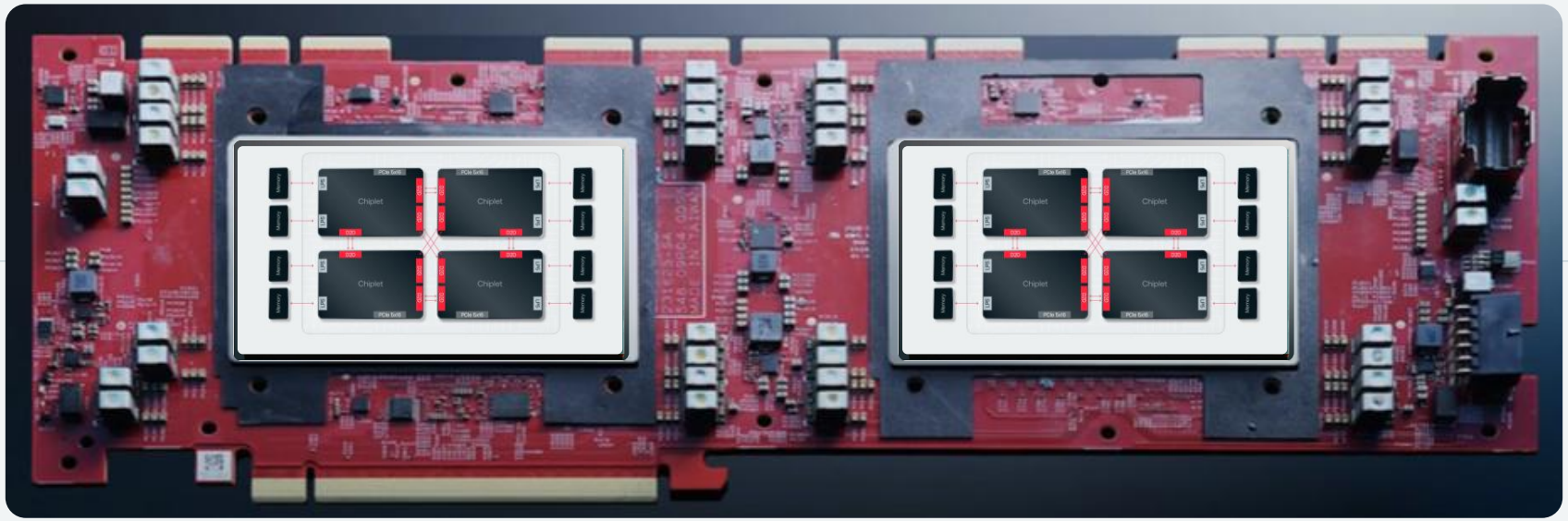
d-Matrix

# d-Matrix Corsair: Chiplet-based Inference Acceleration Platform

8 Chiplets
TSMC 6nm

2400 TFLOPS MXINT8
9600 TFOPS MXINT4

PCIe: 128 GB/s
DMX Bridge: 512 GB/s

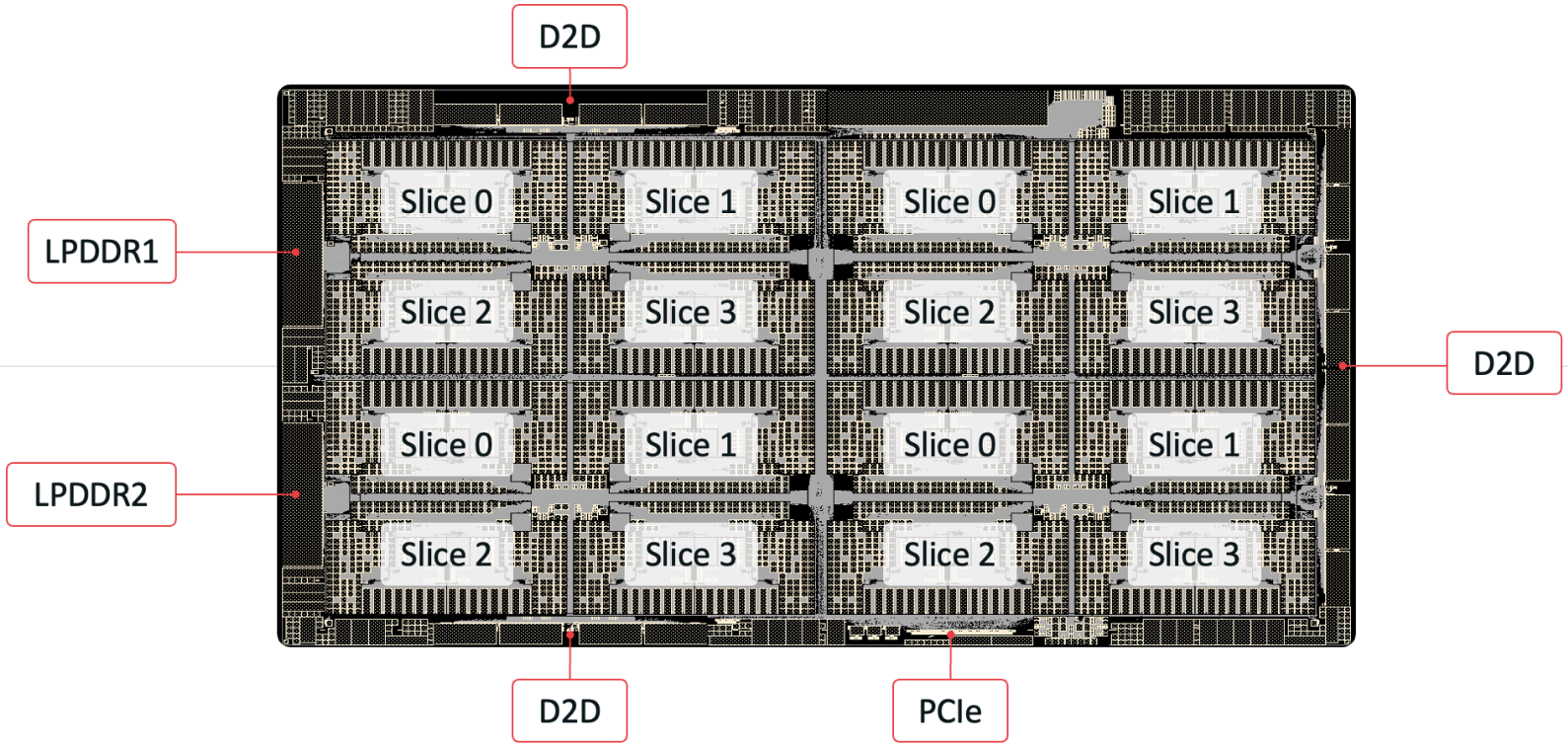


PCIe Gen5
600W TDP

2GB SRAM
Memory 150 TB/s

256GB LPDDR5
400 GB/s

# Corsair Chiplet




THEME ARTICLE: CONTEMPORARY INDUSTRY PRODUCTS

**Corsair: An In-Memory Computing Chiplet Architecture for Inference-Time Compute Acceleration**

Satyam Srivastava, Akhil Arunkumar, Nithesh Kurella, Amrit Panda, Gaurav Jain, Purushotham Kamath, Mark Wutzke, Arun Tiruvur, Mike Gupta, Ilya Soloveychik, Vamsi Darsi, Malav Dalal, Vinayak Patankar, Sasidhar Dudyala, Senthil Duraisamy, Santhosh Ramchandran, Raghav Venkatasubramanian, Yuwei Qin, Xin Wang, Jayaprakash Balachandran, Ali Murat Gok, Piotr Wojciechowski, Saliya Ekanayake, Chris Ng, Ranju Sarma, Shubhankit Rathore, Tristan Trouwen, Siwei Zhuang, Chris Nicol, and Sudeep Bhoja,
*d-Matrix Corporation, Santa Clara, CA, 95054, USA*

*Advances in generative AI (GenAI) have reinvigorated research into novel computing architectures such as Transformer. Transformer, characterized by low arithmetic intensity during most of the inference time, has become the cornerstone of GenAI underlying large language models (LLMs) and reasoning models (RMs). Numerous solutions to the intense memory bandwidth problem have been proposed. Corsair is an architecture that targets this need using chiplet design, digital in-memory computing-based matrix engine, efficient die-to-die interconnects, block floating point numerics, and large high-bandwidth on-chip memories. We describe the Corsair chiplet, scaling approaches to compose larger systems, and outline the software stack. We formulate the inference-time requirements of LLM and RM computation, memory bandwidth, memory capacity, and interconnect efficiency for scaling. We also show how the Corsair design fits perfectly these workloads. We present benchmark results from Corsair silicon that correlate strongly with the design and preview an estimate of workload-level improvements expected with Corsair.*

Large language models (LLMs) have become the cornerstone of modern artificial intelligence (AI). These models have traditionally charted the path indicated by the LLM scaling laws with state-of-the-art models sporting hundreds of billions or even trillions of parameters.[1,2] However, after reaching the trillion parameter scale in 2023, the scaling laws of LLM pre-training seem to have plateaued. This is because 1) the computational needs of training ever larger models are proving to be impractical, and 2) the available data to train the models is finite. Today, we are at the onset of a new paradigm; that of inference-time compute where relatively smaller models achieve higher accuracy by allowing for iteration and reasoning during inference.

While reasoning models[3,4] achieve superior results on various measures of model quality, they come with significant performance overheads. For example, a one billion (1B) parameter reasoning LLM that uses up to 128 generations, achieves a math500 score similar to an 8 billion parameter model under zero-shot inference. However, the increased math500 score of the 1B reasoning LLM comes at a considerable performance cost compared to a one billion parameter model under zero-shot inference. Thus, reducing execution latency is crucial for enhancing user experience and lowering deployment costs by enabling higher system throughput.

Four key factors influence LLM inference execution latency. First, the prefill stage is primarily

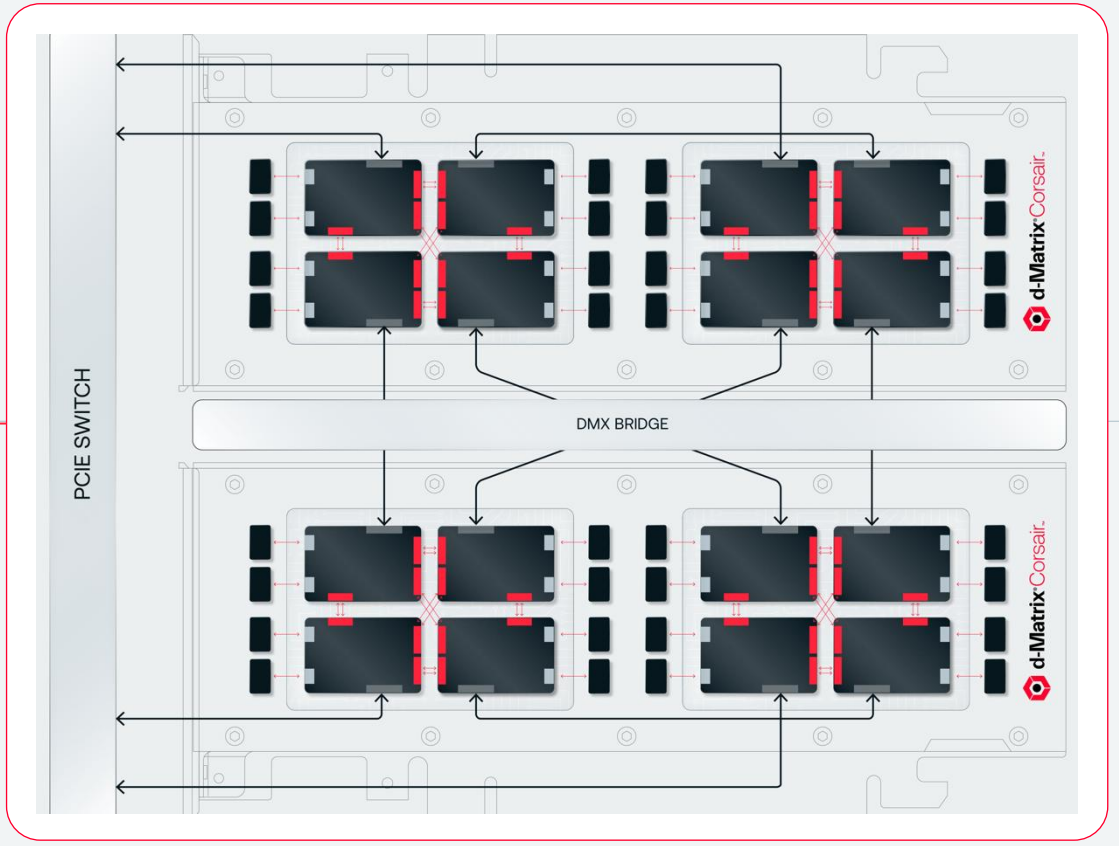

Digital Object Identifier 10.1109/MM.2025.3593444



[Corsair paper: IEEE Micro]

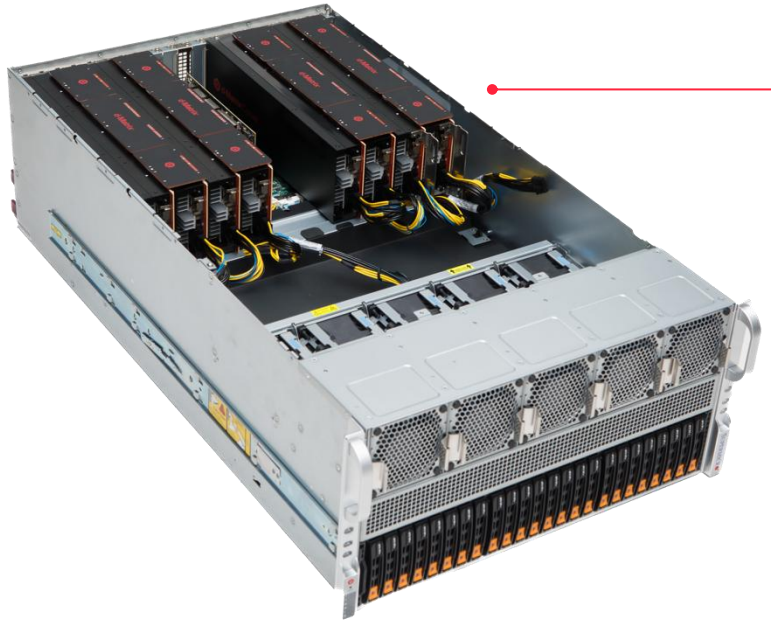
# Card Level Scaleup: 16 Chiplet Hierarchical All-to-All

4 Fully Connected Corsair Packages
Extends Low-latency Collectives
to 16 Chiplets

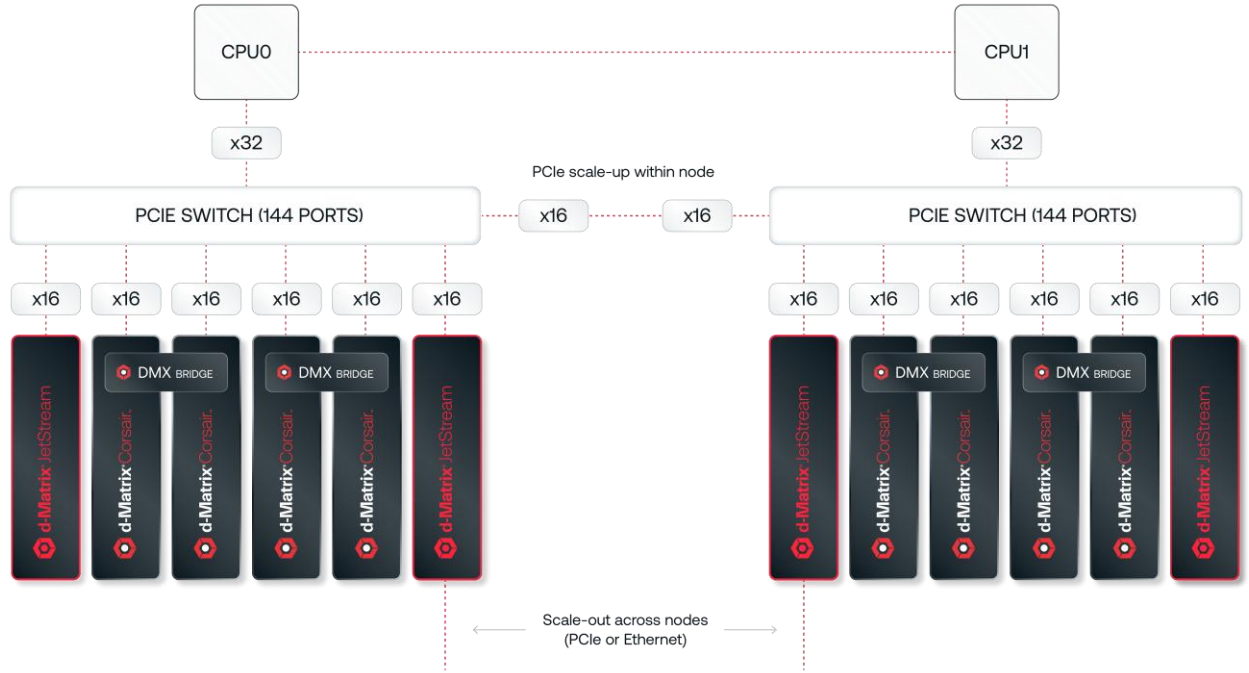



# System and Scaleup Architecture

## Supermicro X14 with Corsair



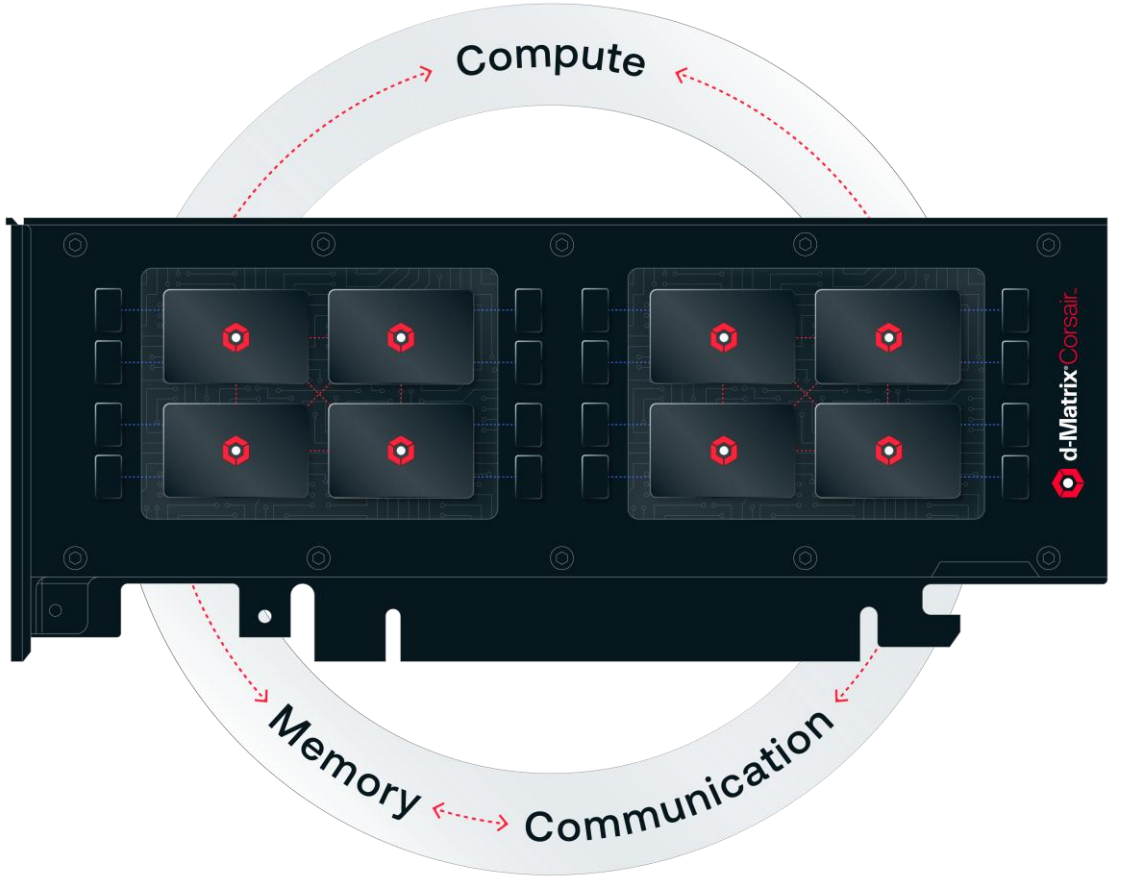
## Server architecture: 16GB SRAM per server




- Composable to rack-level systems
- Scale-up using PCIe within a server: 64 chiplets, switched PCIe
- Scale-out using Transparent NIC (TNIC)

# Corsair Key Pillars – Low Latency, Batched Throughput

- Digital In-Memory Compute (DIMC):
  - Efficiency: 38 TOPS/W
  - 2400-9600 TOPS (eff 8-bit/4-bit precision)
- Block-Floating Point Numerics
  - Accuracy at low-precisions
  - Standardized in OCP MX specs
- 2GB SRAM, 150 TB/s Bandwidth per PCIe
- Low Latency High Bandwidth Interconnects
  - 16 chiplets hierarchical all2all
    - 4x chiplets per package, 4x Packages

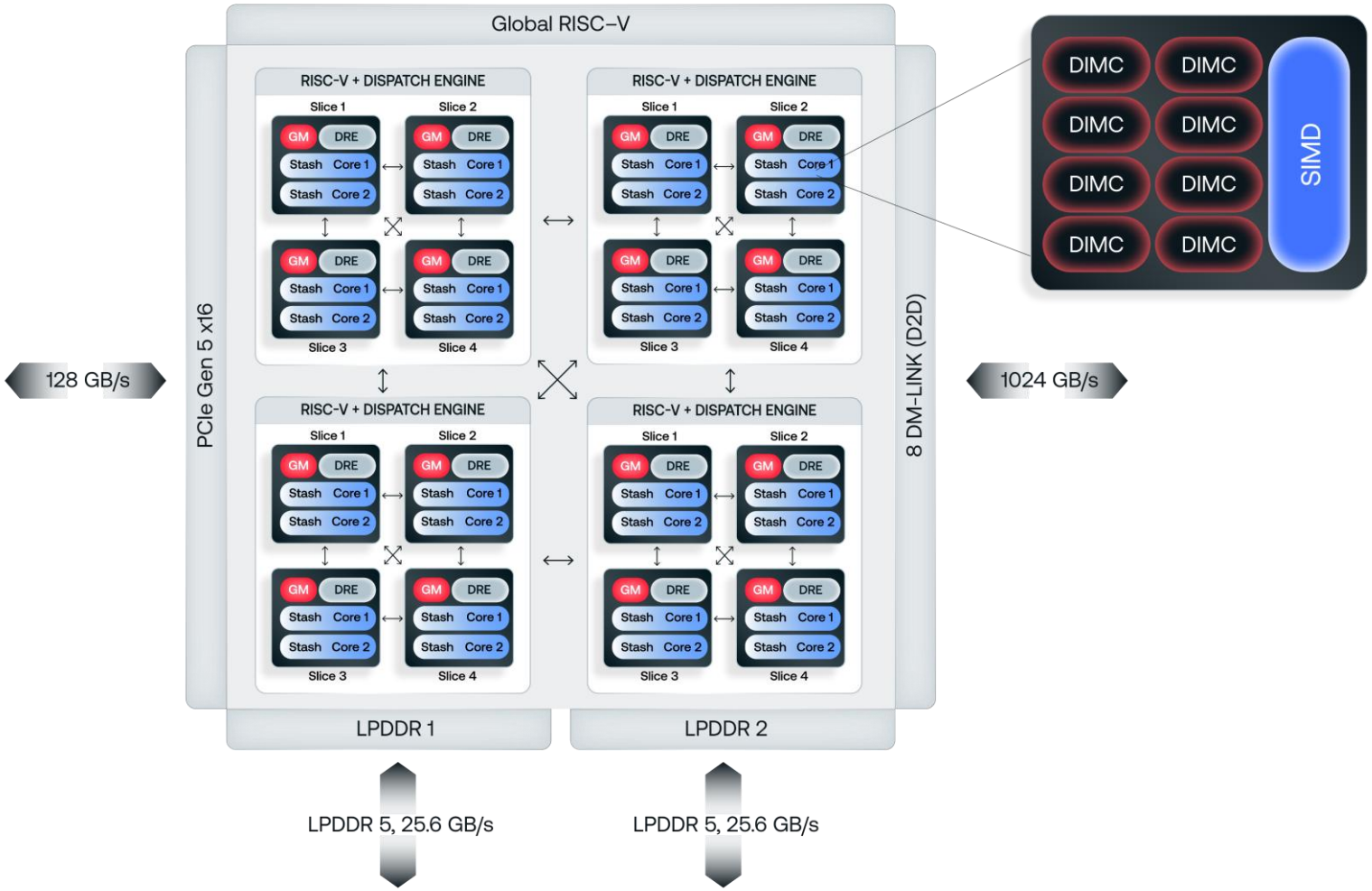


d-Matrix

# Outline

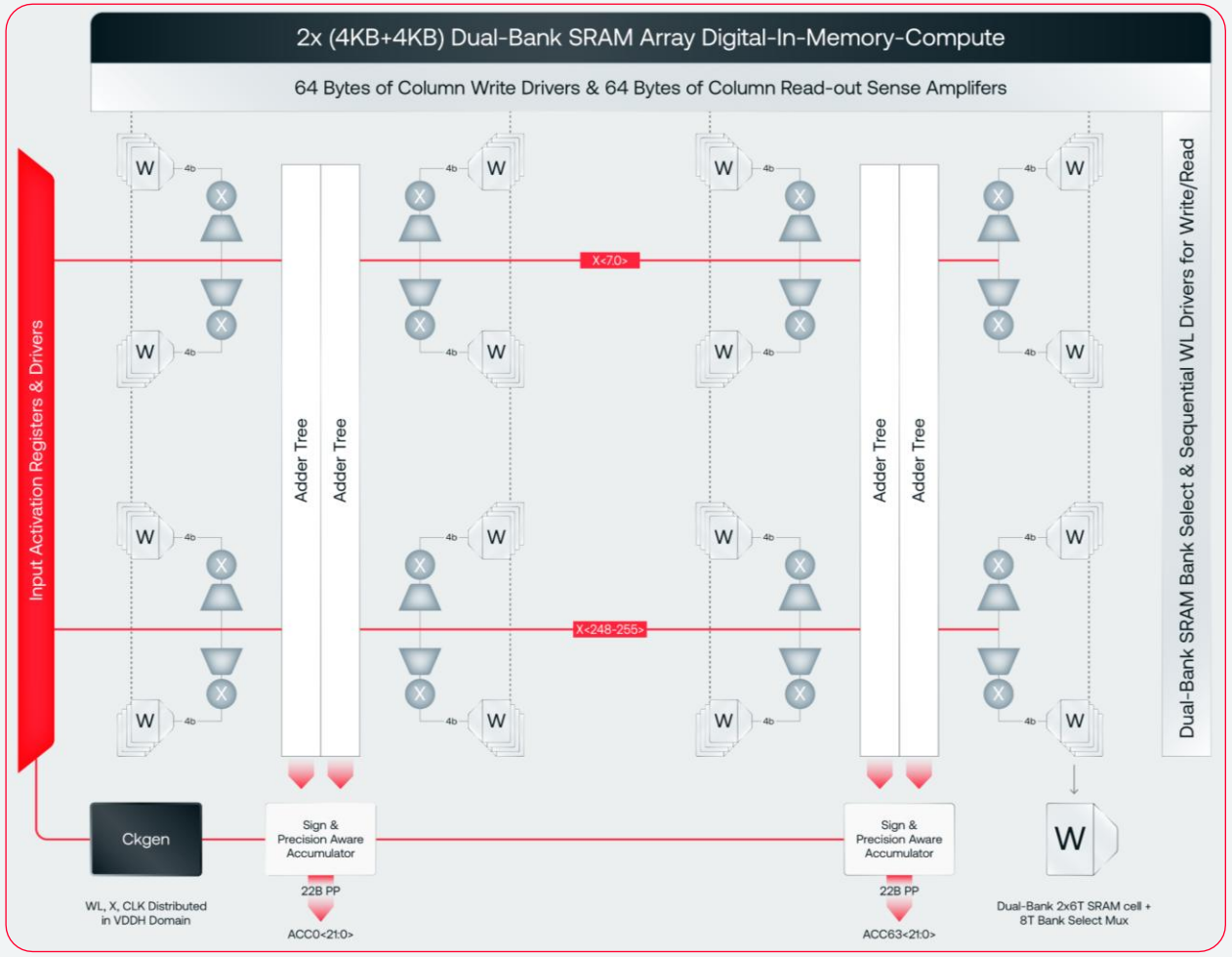
- **Chiplet**
  - **Compute System** → Core | DIMC | Numerics | Compression
  - **Memory System** → Global Memory (GM) | Stash | LPDDR
- **Network**
  - **Hybrid Parallelism**: Challenges and solutions
  - **Connecting Chiplets**: Die-to-Die Links
  - **Connecting Cards**: Scale Up
  - **Scaling Out** for Larger Models → Streaming NICs | Racks
- **Aviator Software Stack**
- **Performance**
- **Future Directions**

# Corsair Chiplet Built Using Modular Hardware Blocks



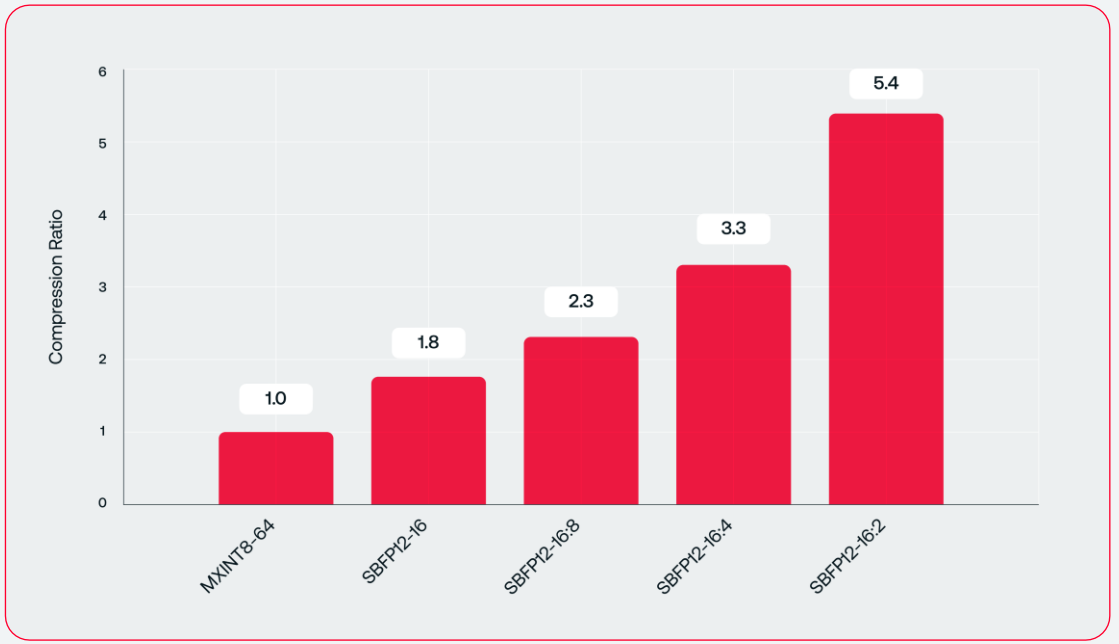

- Corsair SOC Chiplet
  - 4 Quads
  - Global RISC V control core
  - LPDDR & PCIe subsystems
  - D2D
- Quad
  - Locus of control, RISC-V + DE
  - Ease of Programmability
  - 4 Slices clustered with a RISC V control core
  - Instruction dispatcher (DE)
- Slice
  - 2x Cores sharing a global memory (GM)
  - Data reshaper, low latency NOC

# Energy Efficient DIMC Architecture



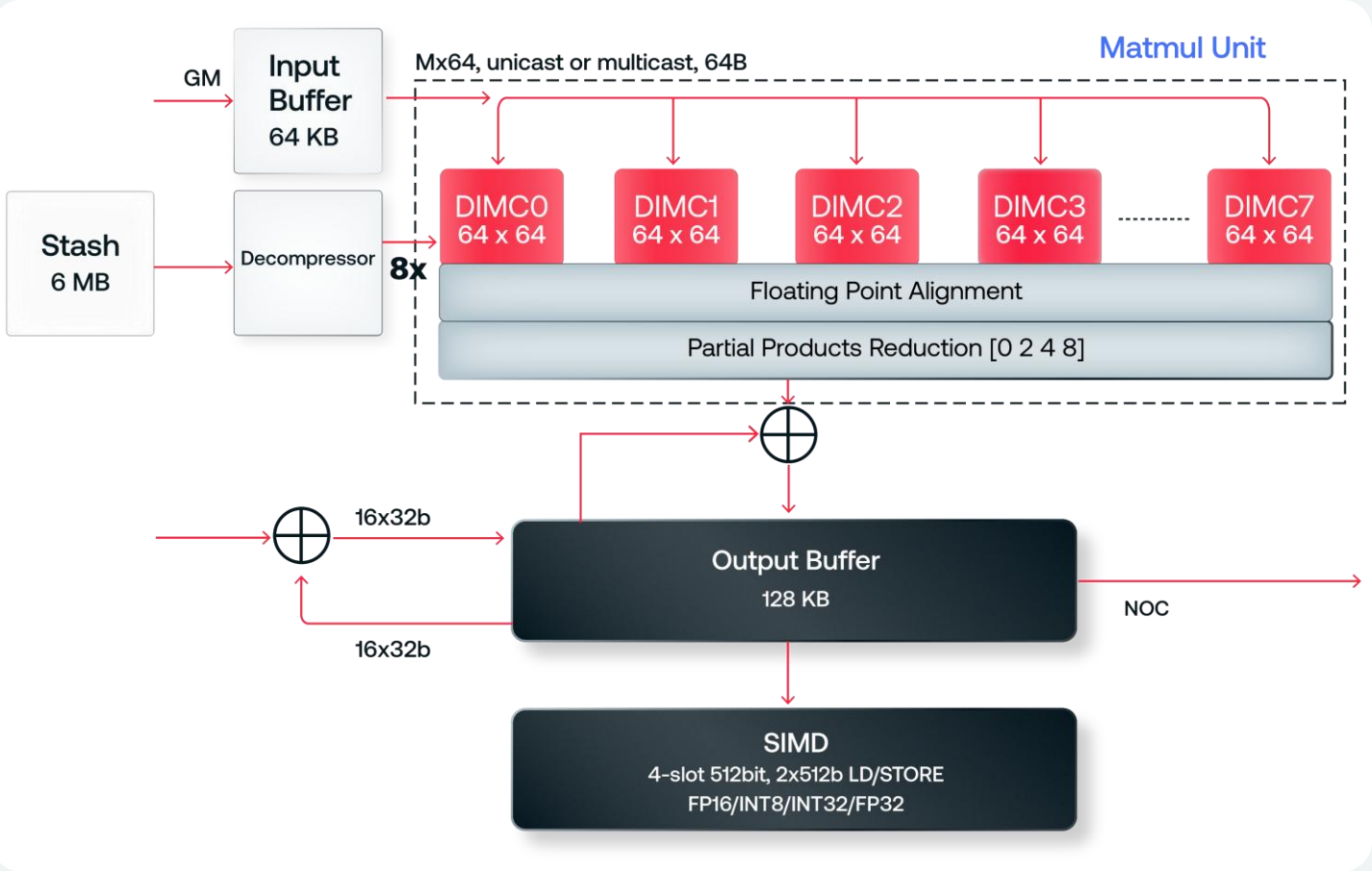

- Logic-rule SRAM array in TSMC 6nm
- Dual bit-serial, 4 bit-parallel MAC array
- Supports 64x64 Matrix Ops in MXINT8
- Supports 64x128 Matrix Ops in MXINT4
- Fully digital design for accuracy and precision

# Corsair supports 5x Weight Compression

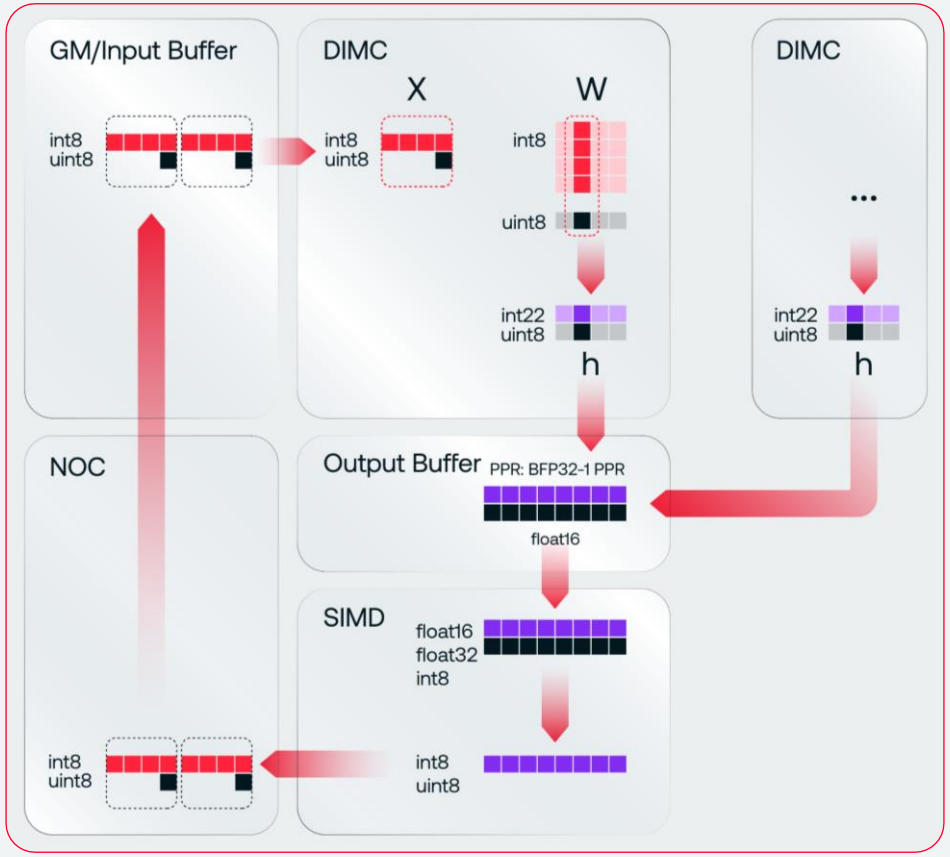

Storage formats with line rate decompression

# Core Architecture



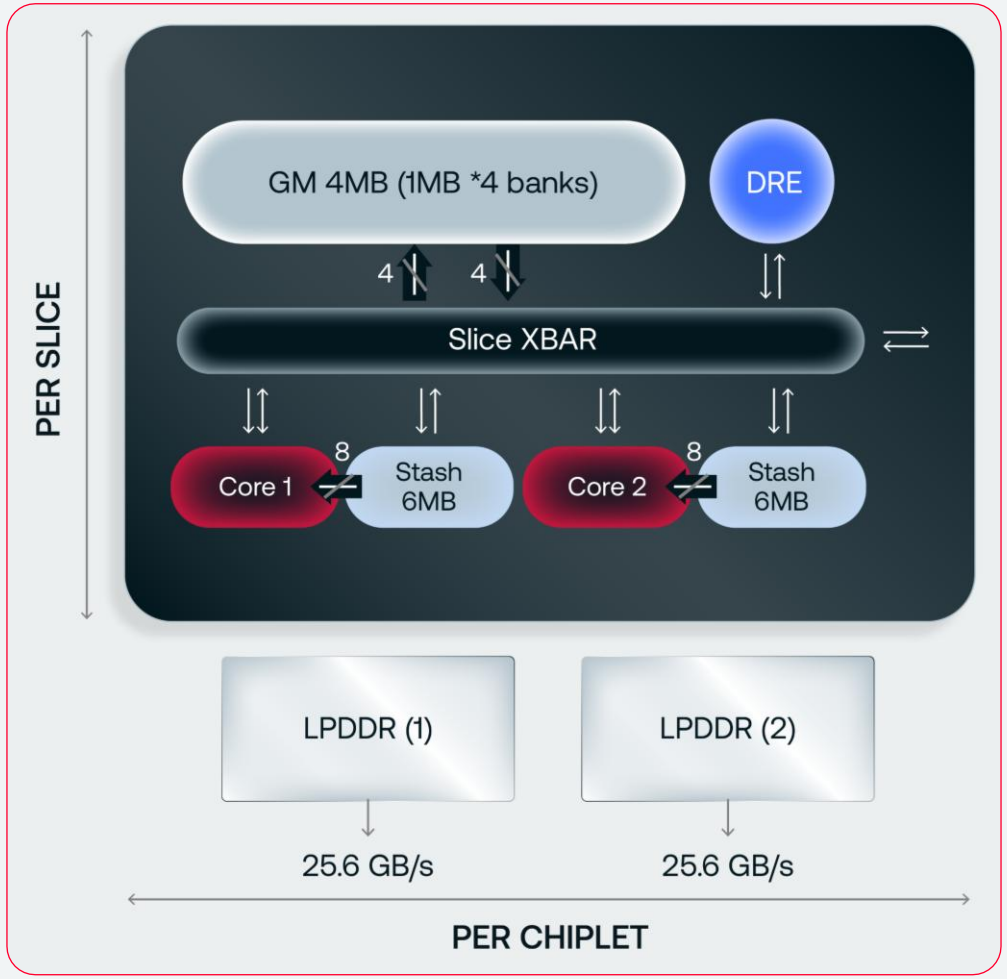

- MX block floating point datatypes
- DIMC array with high BW Stash SRAM
- Inline partial product reduction
- 4 wide VLIW Vector SIMD unit
- On-the-fly quantization/format conversions

# Dataflow with Block Floating-Point Numerical Formats




Integer MAC efficiency with floating point accuracy

# Memory System: Global Memory, Stash, and LPDDR




- **Stash**: Where weights are stored
- **Global Memory** (GM): Where activations are stored
- Stash and Global memory: Vector (64 bytes) addressable
  - GM BW = 307 GB/s, Stash BW = 614 GB/s
- DMA
  - Transfer contiguous and/or strided data with looping
  - 4D Nested loops
- Data Reshape Engine (DRE)
  - Transpose, tensor insertion, and tensor extraction
- **LPDDR** – Per Chiplet
  - Two interfaces: 51.2GB/s (aggregate)
  - Prefill-Decode disaggregation, Prefix caching, dormant KV cache

d-Matrix

# Scaling Challenges for Large-Model Inference

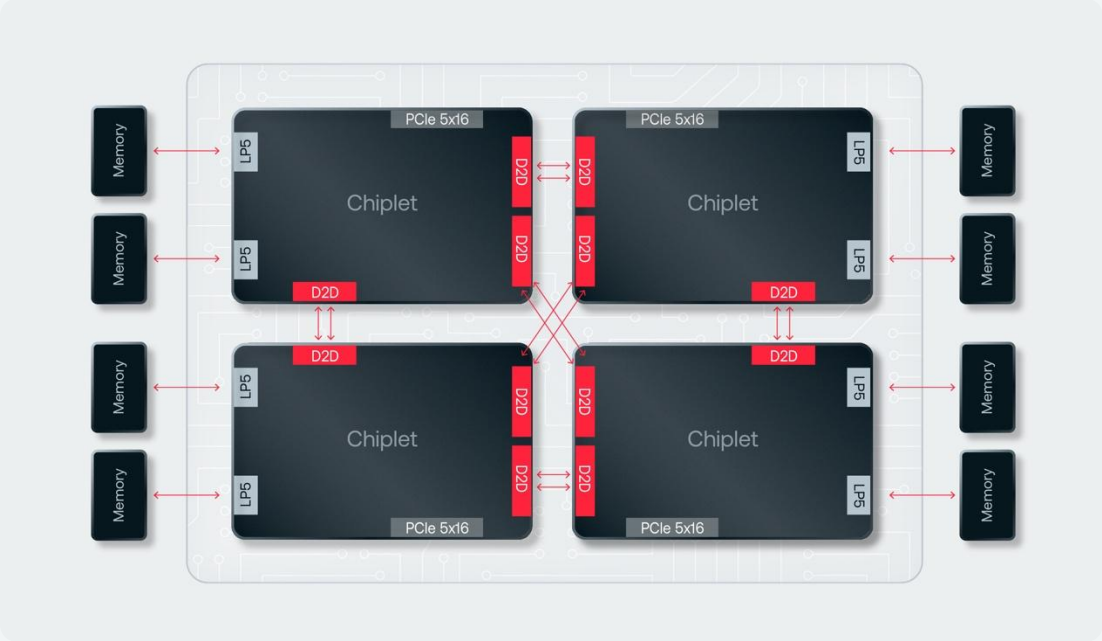
- Model Parallelism -> Hybrid Parallelism
  - Aggressive tensor, pipeline, expert, and sequence parallelism
- Distinct workload phases with varying demands from the system
  - Prompt processing (BW bound communication)
  - Token generation (tight-latency bound communication)
- All-to-all patterns in SOTA workloads
  - Hybrid parallelism inserts many different collectives (All-Reduce, All-Gather, All-to-All, Reduce-Scatter, etc.)
  - Typically blocking and latency–dominated.
- Collective latency is drowning out FLOPS
- Fully-connected network topologies are hard to scale
  - Physical fully-connected topology – cost of wires, routing etc.
  - Logical fully-connected topology (over switch) – Port bifurcation at devices, high radix switches

# Corsair Scaleup – Hardware-Software Codesign

- Models are typically sharded on Corsair using:
  - Tensor Parallelism - scaleup
  - Pipeline Parallelism – scaleout
  - Expert Parallelism - scaleout
- Corsair scaleup
  - Hardware – high bandwidth, low latency and fully-connected topology
    - 16 chiplets hierarchical all2all, dual card, 4 packages directly bridged with 128GB/s pairwise BW
    - 128GB/s pairwise D2D, bridged PCIe, switched PCIe connections
    - 115ns D2D, Bridge PCIe 400ns, Switches PCIe 650ns
    - 64 chiplets, 8 cards, switched PCIe
  - Software – one-hop tree-based collectives
- Corsair efficiently supports both:
  - High bandwidth: large tensors (prompt processing)
  - Low Latency: small tensors (token generation)

d-Matrix

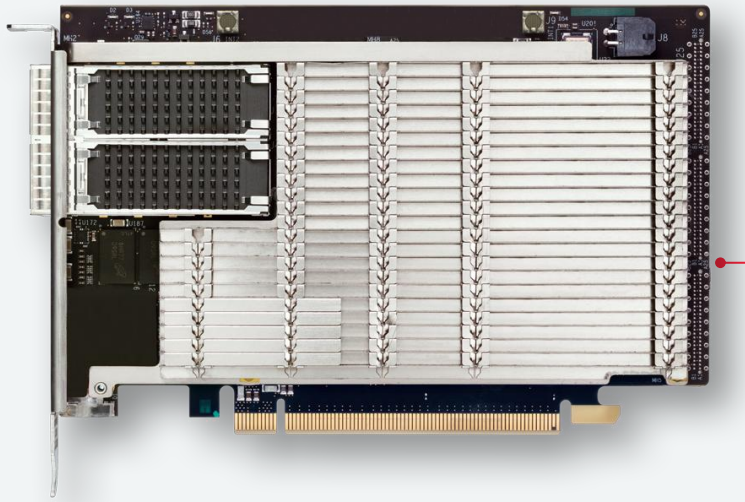
# Package Level Scaleup: 4 Chiplet All-to-All

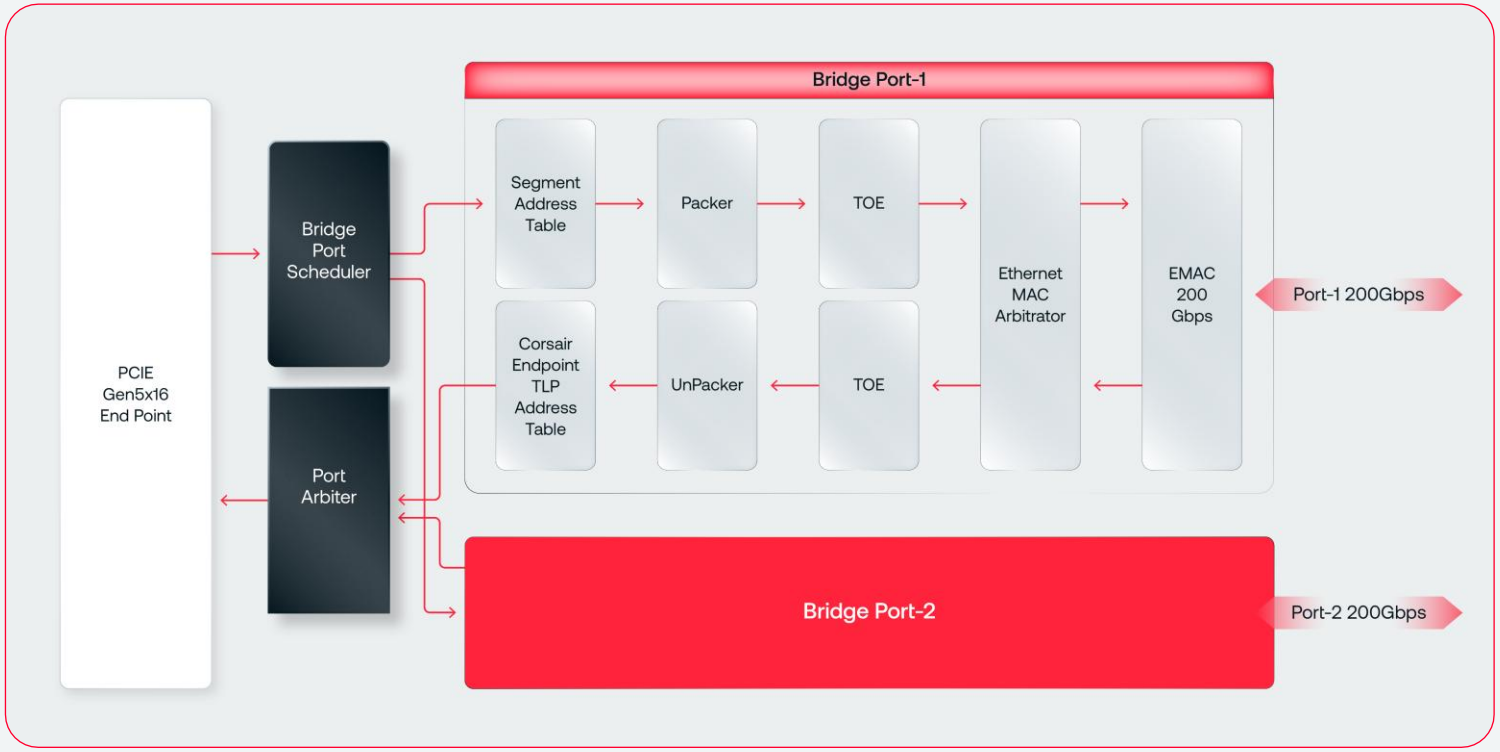




- 4 chiplets all-to-all
- Pairwise BW of 128 GB/s (x2 D2D links)
- 16G, 0.35pJ/bit, RS FEC

# Transparent NIC – Ethernet Scale-Out

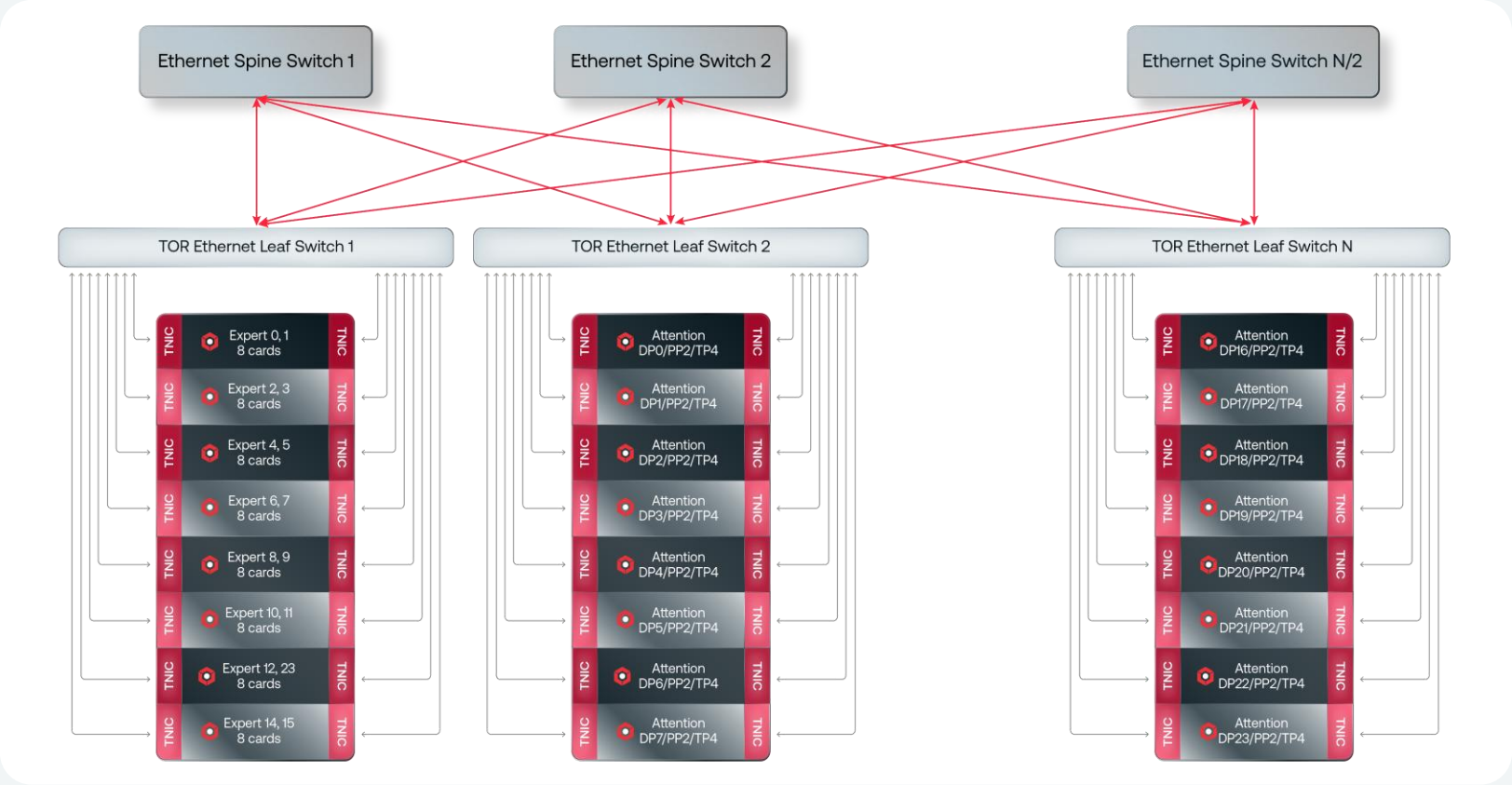





- Transport PCIE semantics over Ethernet
- Scale communication semantics from intra-chiplet to inter-node
- Lower latency 2us vs RDMA 4us

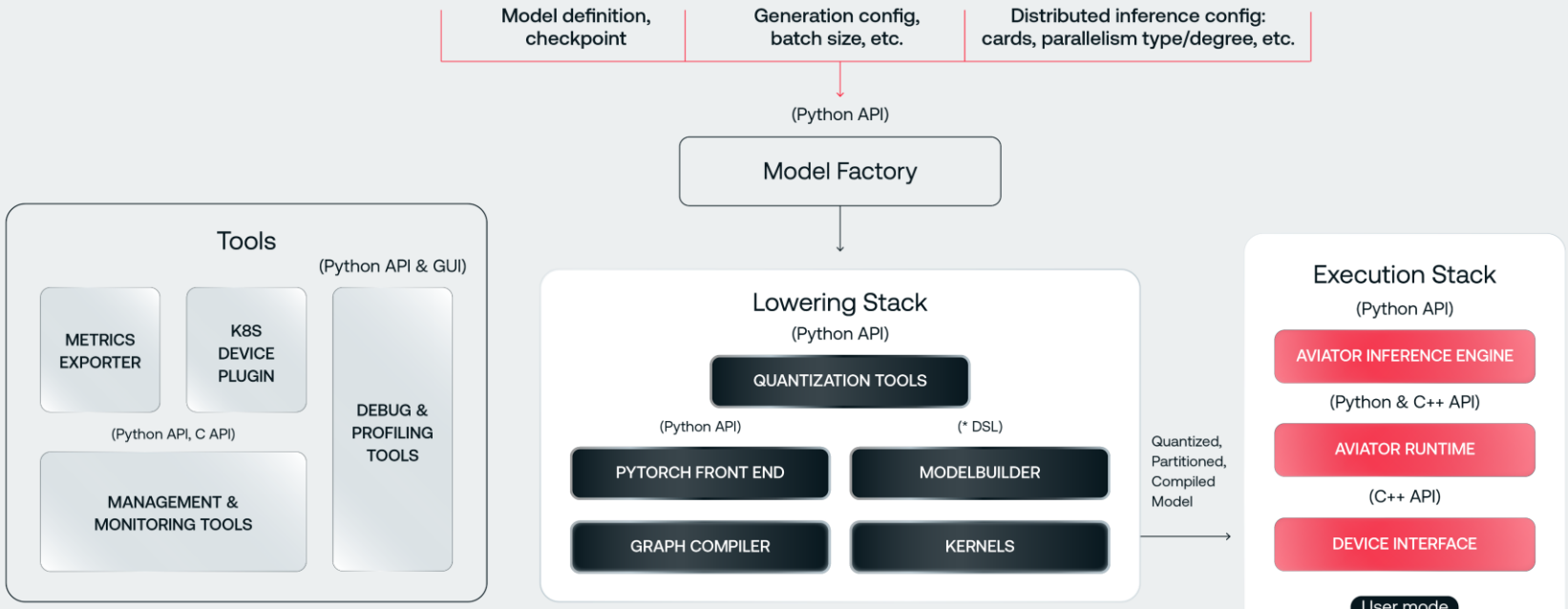
d-Matrix

# Rack Level Scale-Out: Multi-node and Multi-Rack




Supports range of models including MoE

# Aviator Software: Easy to use and optimized for Corsair

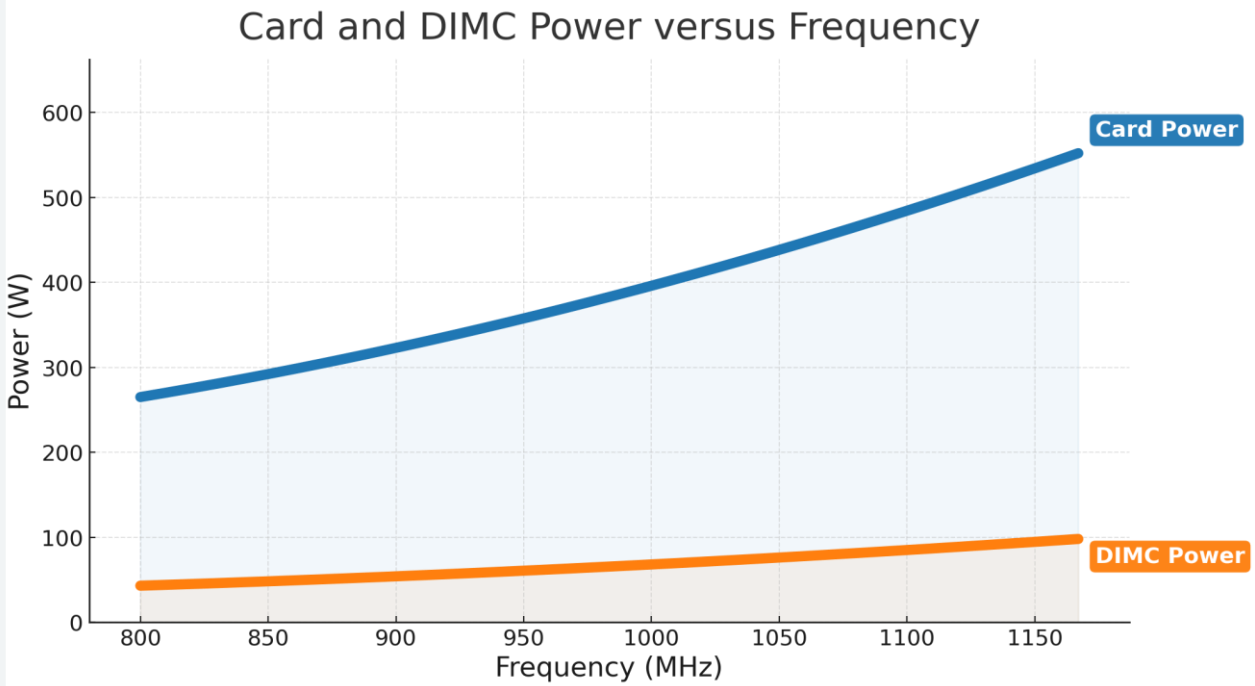


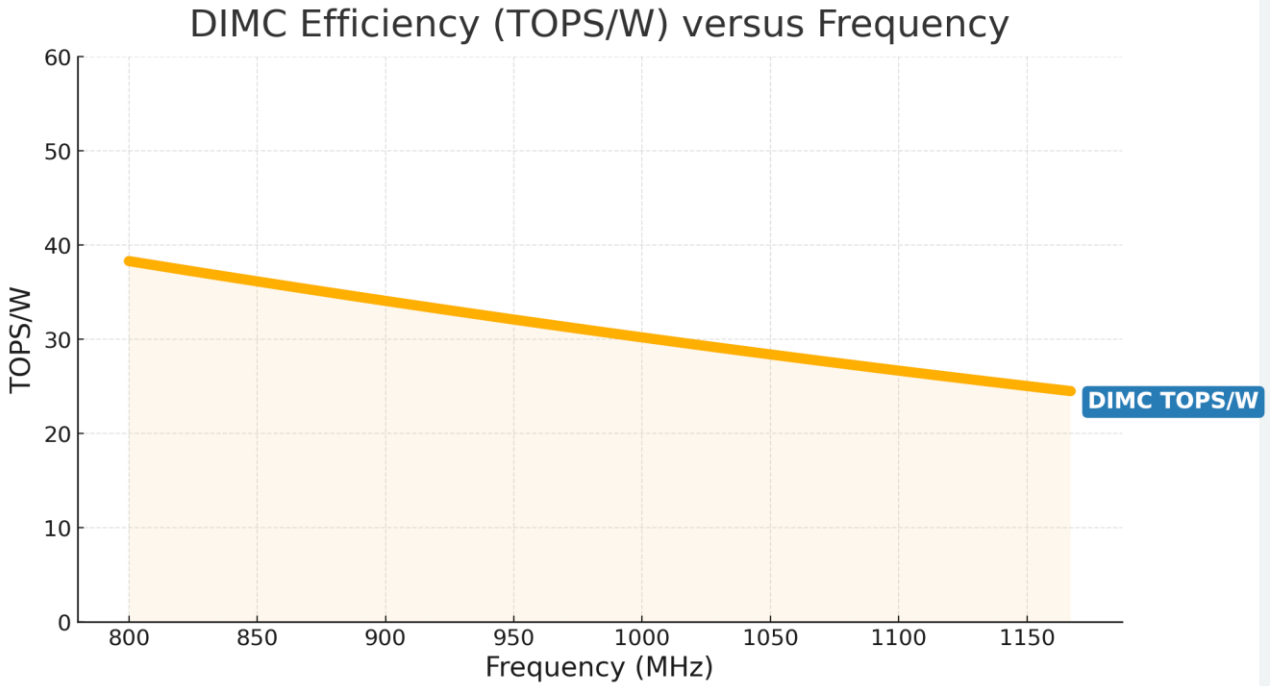

- Familiar model authoring and ingestion interface with GPU-like parallel ops
- Consume pre-trained weights using push-button quantization tools
- Topology-aware collectives embedded in the exec graph, run with no host involvement

# Aviator Software: Codesigned for LLM Acceleration

| Software Feature | Enabled By |
|---|---|
| Minimal host-device sync | Continuous and asynchronous graph exec, check for new work |
| Low latency collectives | Graph-native invocation with sync hints |
| Expert score and routing | Native support for conditions, dynamic addresses |
| Iterative token generation and dynamic shapes | On-device autoregression engine |
| Efficient LLM Serving Engine | Minimized IPC, O(1) critical-path host-processing with card count |

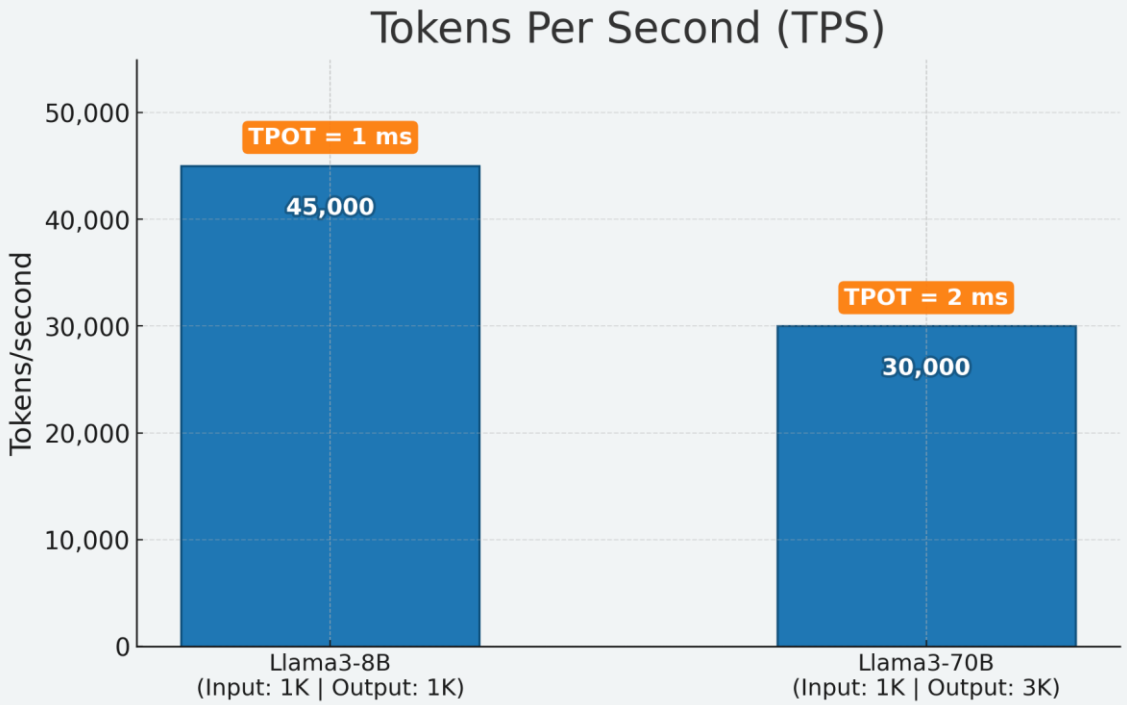
# Power Efficiency (TOPS/W)




Measured for power stress → Worst-Case Power Pattern.
Workload power: Expected to be 60%-80% of the reported power.
Card power includes SIMD, XBARs, IO, DDR, regulator power loss, etc.

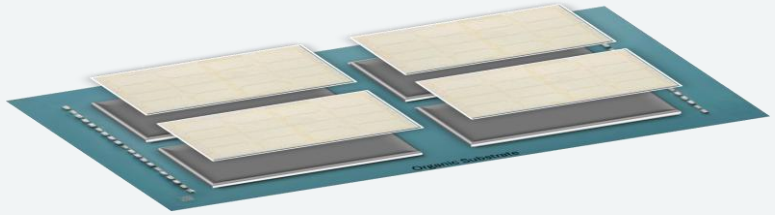
# Performance and Flexibility for Use Cases



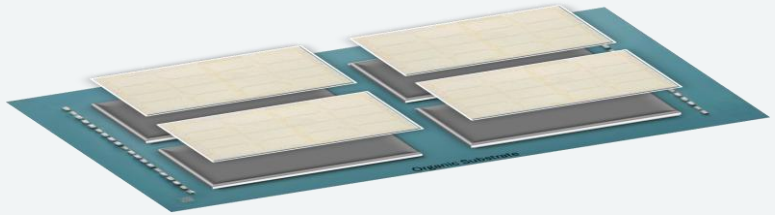


Performance numbers are preliminary and subject to change. Results may vary.

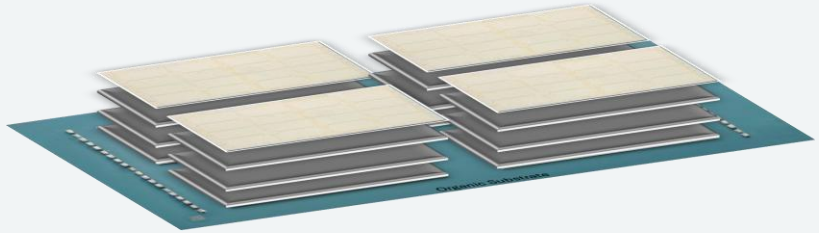
# Stacking of logic on DRAM interposer in 3D



Corsair with Passive Silicon capacitor interposer



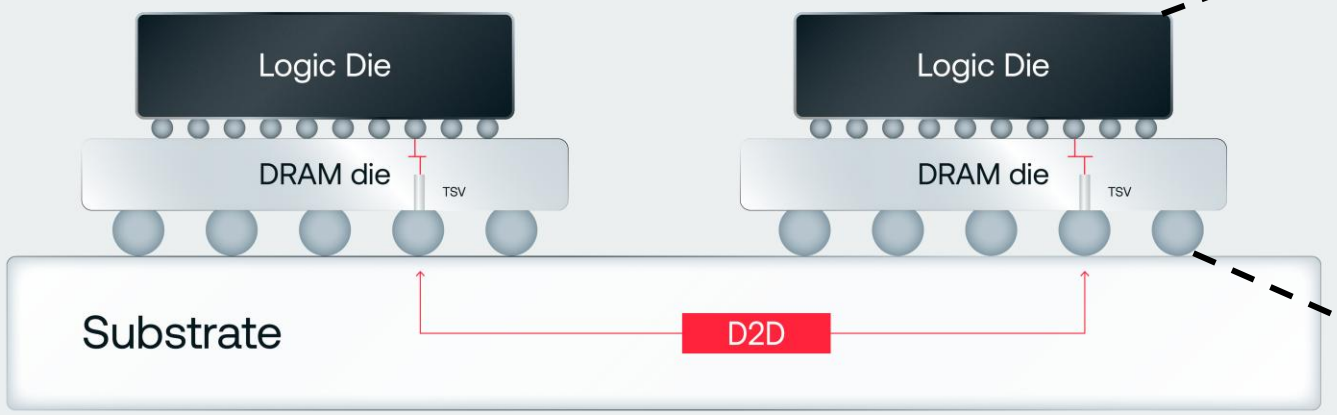
3D- DRAM Single Stack

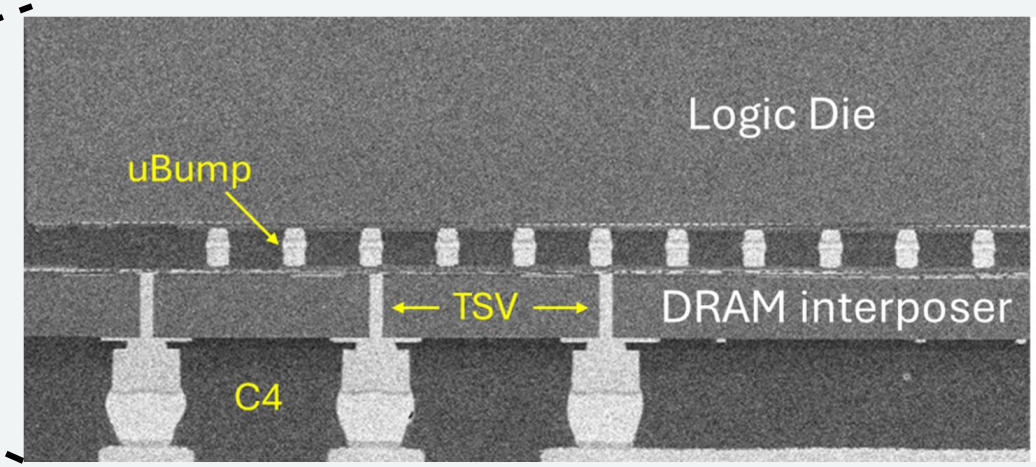


3D- DRAM Multi-Stacked

- Passive DRAM based silicon capacitor interposer improves power integrity in Corsair
- Technology is adapted into an active DRAM interposer to create a single-stack 3D DRAM
- Increases memory capacity by orders of magnitude
- 3D DRAM exposing small banks
- 100x vertical connections direct to compute relative to HBM4
- 10x BW & energy efficiency per stack
  - HBM4 : 2TB/s, ~3pJ/bit + additional energy on XPU die
  - 3DIMC: 20TB/s, 0.3pJ/bit

# 3D DRAM Test Vehicle




- Top die: TSMC N5 logic
- Bottom die: 3D DRAM
- Integration: 36µ Face to Face (F2F) stacking
- Proven low cost, high volume, high yield process

d-Matrix

# Thank You

Contact:

Sudeep Bhoja

sbhoja@d-matrix.ai

d-Matrix

# Backup

d-Matrix

# Tying it together

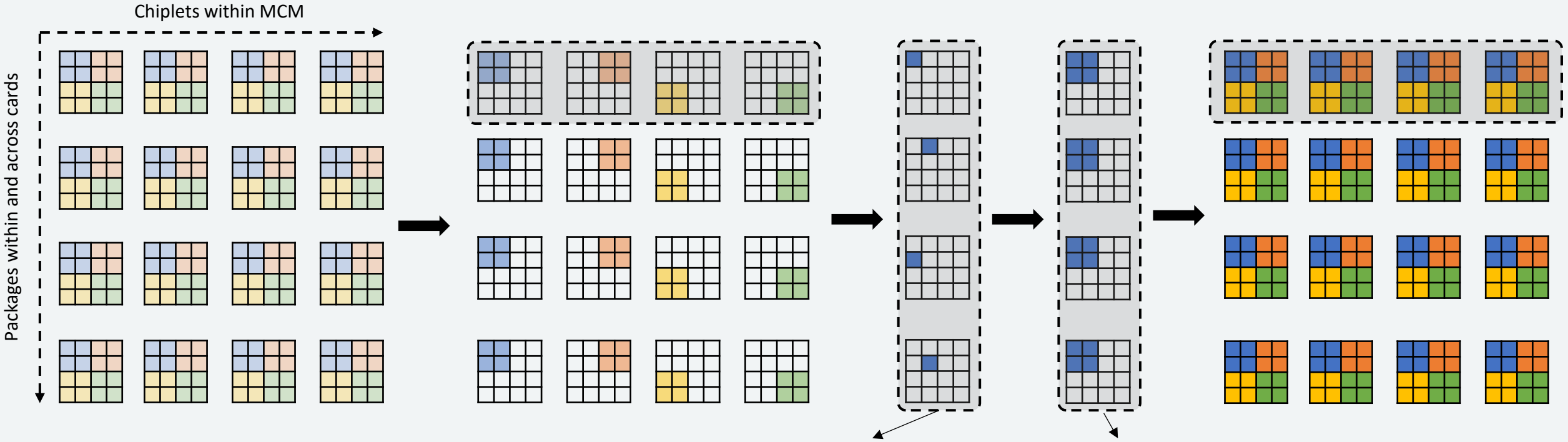
- Workloads/Kernels Card-level
- High-Level Mem: LPDDR
- Slice-Level Mem: GM
- Instructions are run at Cores
- Low-Level Mem: IB/OB/WB/ST Mem

# Aviator: Low-Latency In-Graph Collectives

Initial state: each chiplet has tensor of size “T” (partial product of a GEMM)

#1 - **Reduce-Scatter (D2D)** between 4 chiplets in a MCM
All MCMs work independently
Data moved between pair of chiplets – T/4

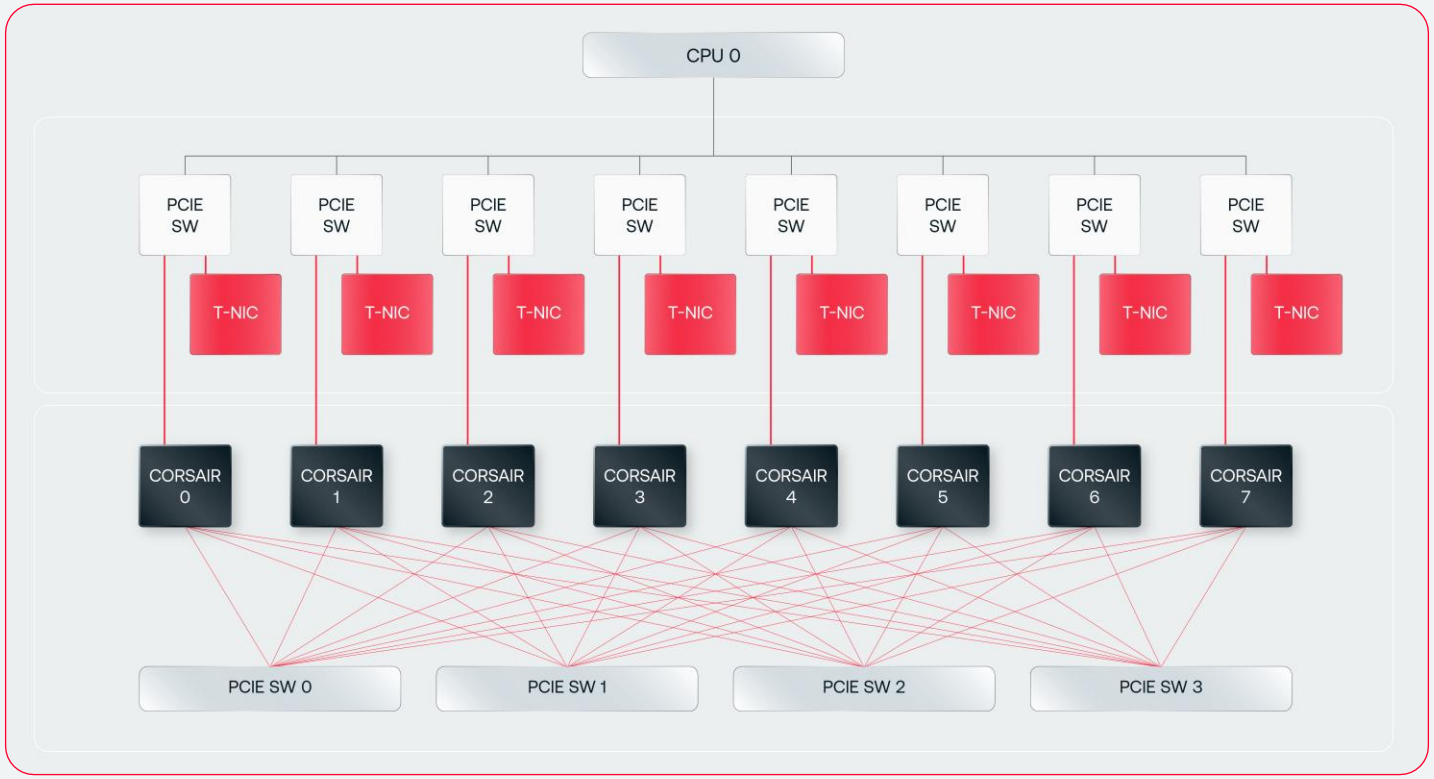
#3 - **AllGather (D2D)** between 4 chiplets in a MCM
All MCMs work independently
Data moved between pair of chiplets – T/4




#2a - **Reduce-Scatter (PCIe)** between 4 chiplets across 4 MCMs
All chiplets in an MCM work independently
Data moved between pair of chiplets – T/16

#2b – **AllGather (PCIe)** between 4 chiplets across 4 MCMs
All chiplets in an MCM work independently
Data moved between pair of chiplets – T/16

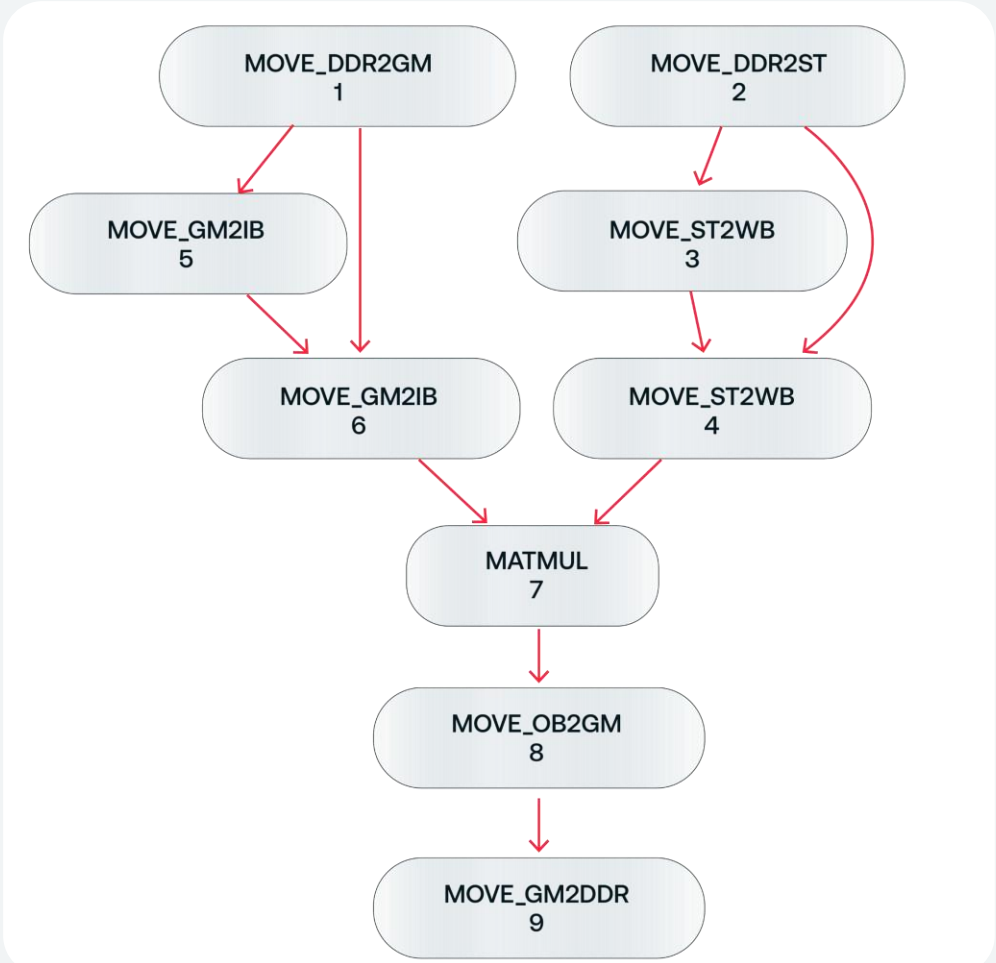
# System Level Scaling to TP8



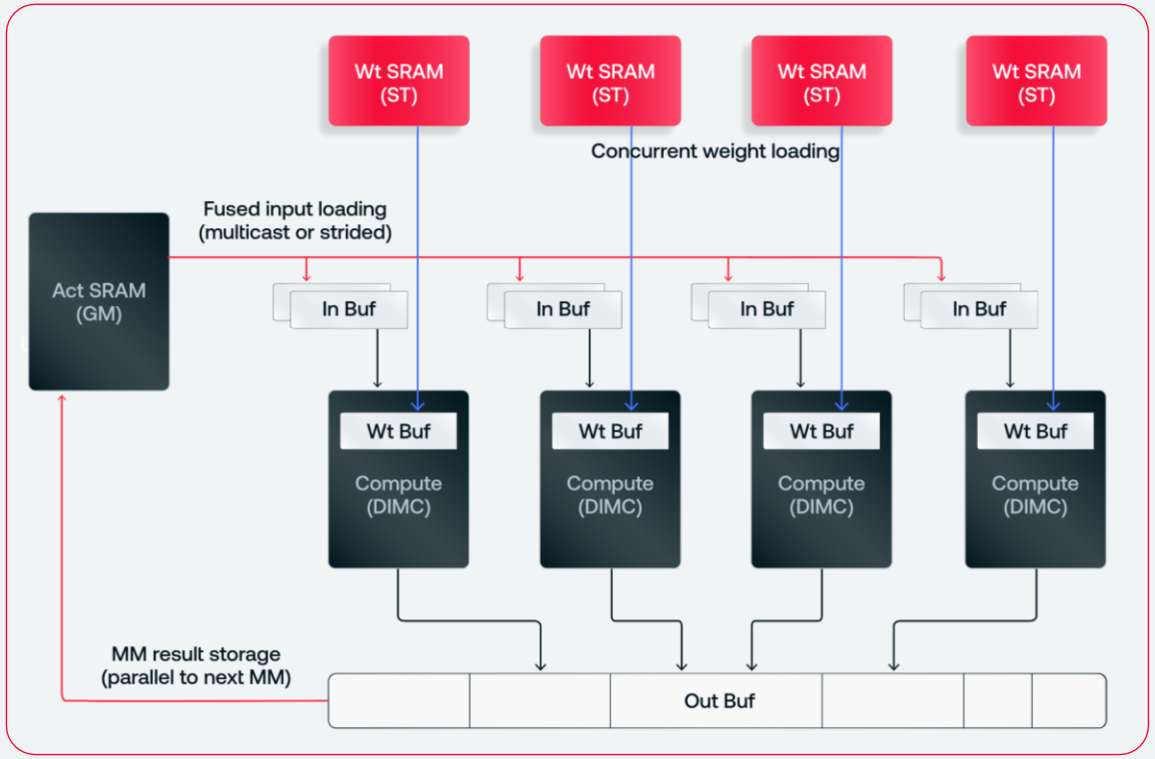

- Larger scale-up domain with more PCIE switches
- Single PCIE switch hierarchy for backend scale-up network
- 1:1 accelerator to NIC for higher scale-out bandwidth

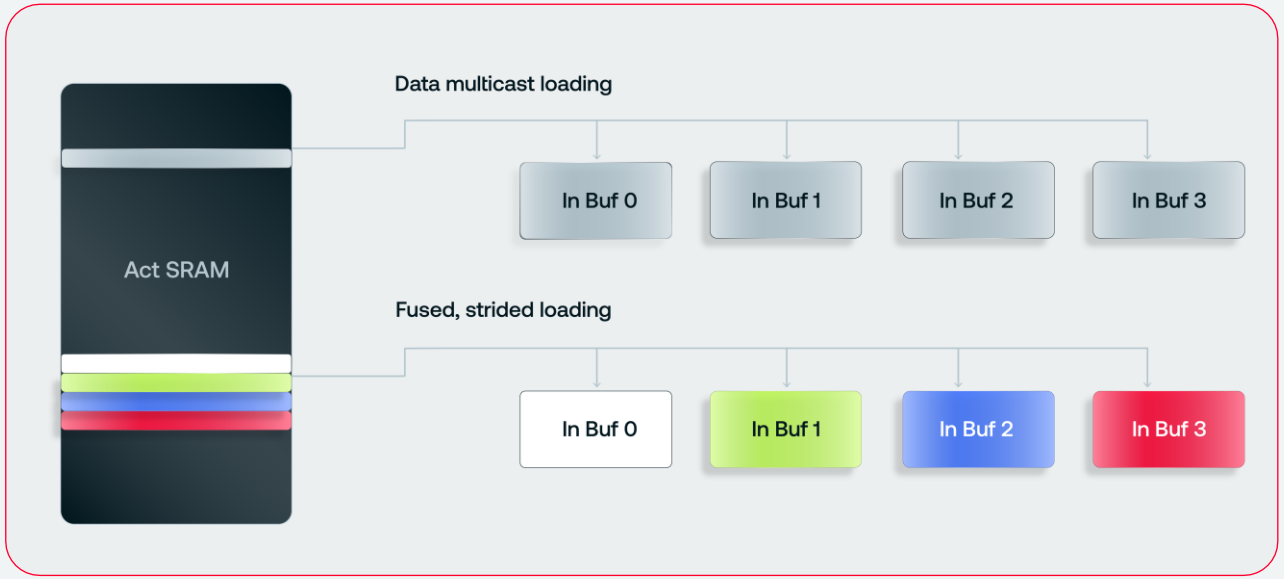
# Aviator: d-Matrix ISA and Graph Representation

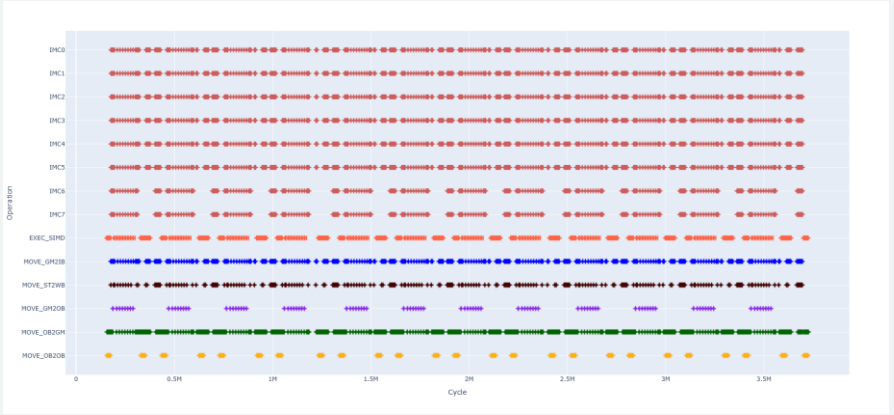
| **Data movement** | |
|---|---|
| MOVE_xxx2yyy | DMA based data transfer from -<br>xxx: DDR, GM, ST, OB<br>yyy: DDR, GM, ST, OB, WB, IB, DRE, SIMD |
| MOVE_xxx2PCIE | PCIe based peer-to-peer transfer |
| **Compute** | |
| MATMUL | Matrix multiplication using DIMC |
| CONV | Direct convolution using DIMC |
| EXEC_SIMD | Kernel function execution on SIMD cores |
| EXEC_CPU | Control function execution using gang CPU |
| **Data reshape** | |
| TRANSPOSE | Array transpose |
| EXTRACT | Rule-based sub-view extraction |
| INSERT | Rule-based sub-tensor insertion |
| **Control** | |
| LOOP | Nestable iterators |
| SEND/REC | CPU messaging via mailboxes |
| CONFIG_zzz | Pre-configure units (AC, SIMD, GM, ST) |
| BARRIER | Various barriers for stalling graph exec |
| RESET | Software triggered reset |
| NOP | Dummy, filler ops used for graph sync |



# Aviator: Performant Matmul Lowering



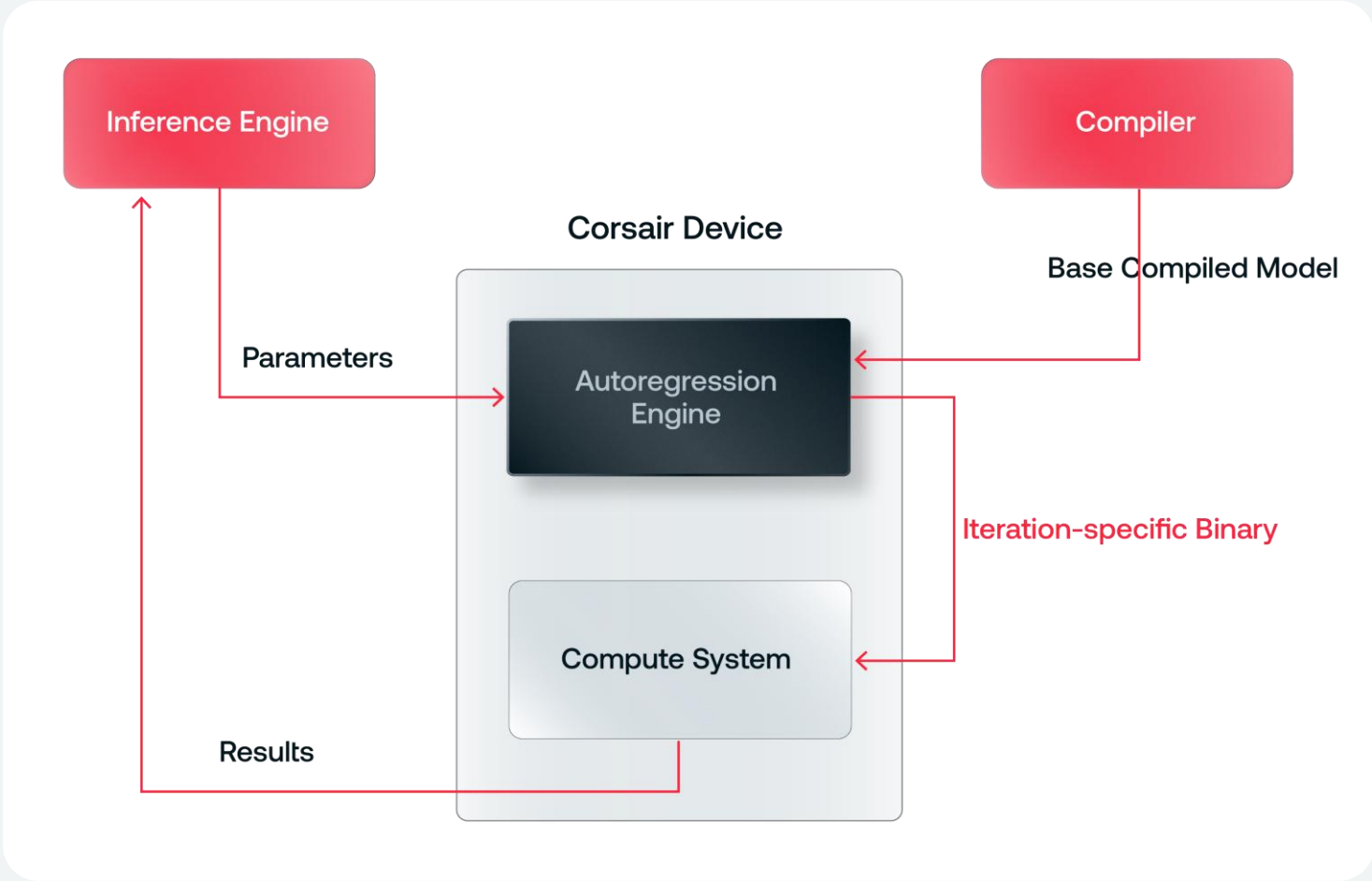



**Performance by <u>sustained</u> DIMC active operation:**

- Pipelined, overlapped computation + communication (ping-pong buffering)
- Minimized data path overhead (data multicast, strided transfers)
- Instruction merging to exploit symmetric patterns (command multicast)

# Aviator: Autoregression

- On device support for autoregressive graph execution
- Auto expands spatial and temporal span with growing context
- No additional host-device interaction
- **No recompilation!**



# Aviator: Inference Engine for LLM Serving

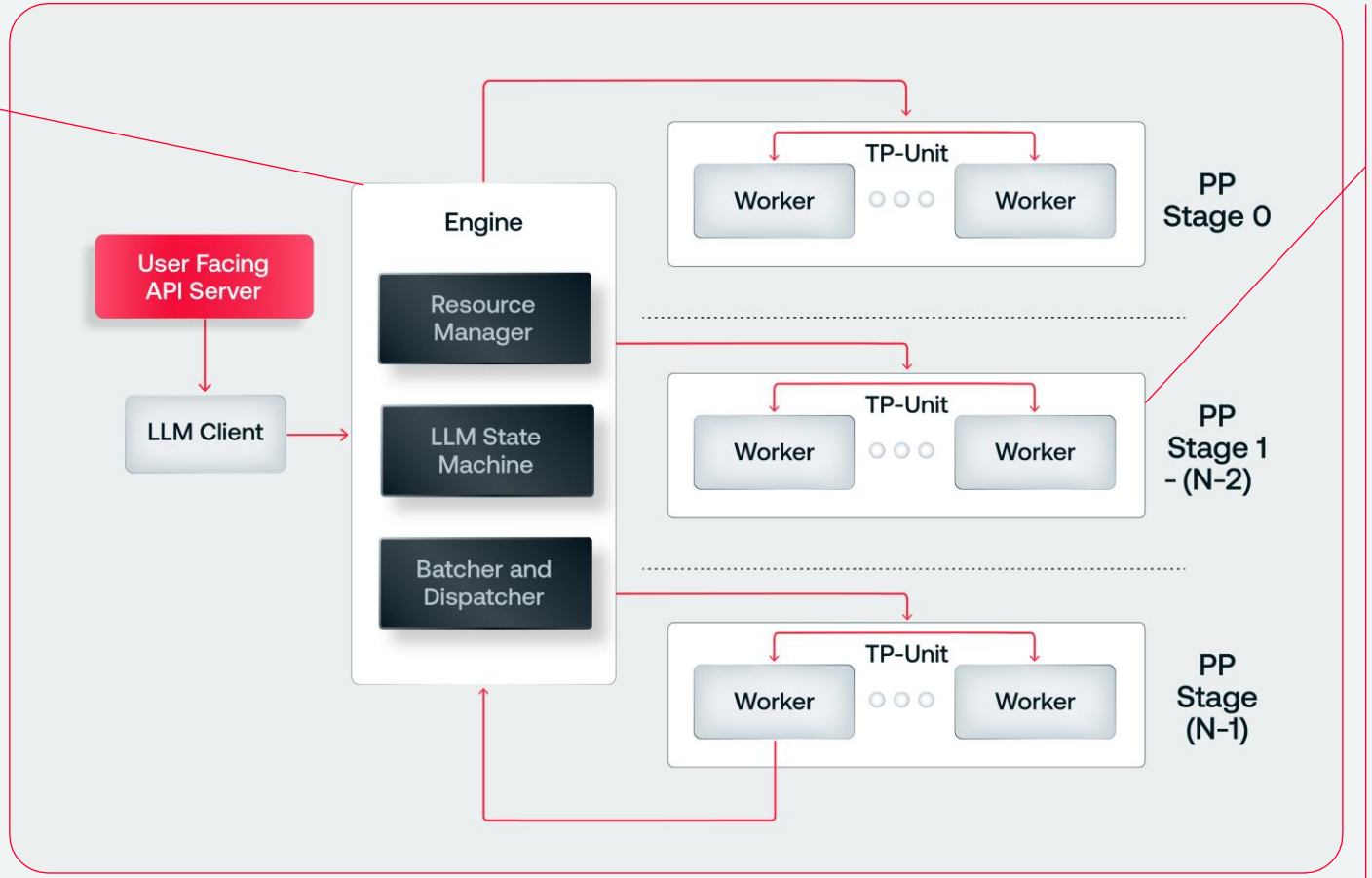
Brain of serving infra

Tensor and Pipeline Parallelism (TP and PP)

Model-aware algos

Dynamic cache mgt

Advanced features (Paged Attention, Continuous Batching, Speculative Decoding)




Distributed, high perf, multi-modal capable inference serving stack

One per card – thin, largely asynchronous

Continuous, async request queue

Minimal worker IPC – none on critical path

Minimal H2D, D2H interaction