DIRECT: Enabling Scalable Processing-In-Memory via DPU-to-DPU Communication

A Hierarchical, CPU-Bypass Architecture with Hardware Synchronization

Prateek P. Kulkarni

PES University, Bengaluru

Email: pkulkarni2425@gmail.com

August 25-26, 2025

6th India ESD Workshop @ IISc

For more, visit my website

Motivation & Problem

- **Scaling ML training** is gated by data movement and synchronization overheads.
- **PIM** reduces CPU↔Memory traffic, but current systems route inter-DPU traffic via CPU.
- Result: **poor scalability**, energy waste, and low parallel efficiency at cluster scale.

Goal: Enable direct DPU-to-DPU communication with hardware-level sync, preserving locality and bypassing CPU.

Baseline Communication Cost (Conventional PIM)

$$T_{\text{comm}} = 2N_{\text{DPU}}\left(L_{\text{DPU-CPU}} + \frac{M}{B_{\text{DPU-CPU}}}\right) + Q_{\text{cont}} + O_{\text{sync}}$$

Where:

- N_{DPU} = number of Data Processing Units
- $L_{\text{DPU-CPU}}$ = DPU-to-CPU communication latency
- M = message size (bytes)
- $B_{\text{DPU-CPU}}$ = available DPU-to-CPU bandwidth
- Q_{cont} = queuing delays from contention
- O_{sync} = synchronization overhead

Key Issues:

- Each transfer pays **latency** + **bandwidth** cost to the CPU.
- Queuing (Q_{cont}) and software barriers (O_{sync}) worsen at scale.

DIRECT: Key Idea & Contributions

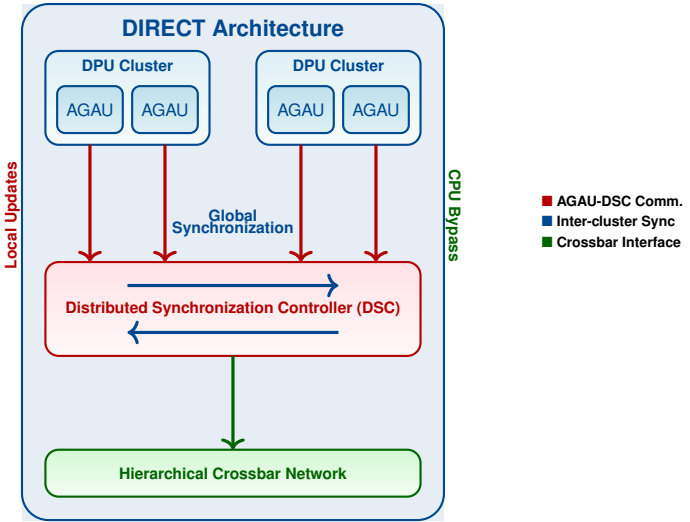
Idea: CPU-free DPU-to-DPU communication via a **hierarchical crossbar**, with **hardware primitives** for local & global sync.

- AGAUs: in-memory **atomic gradient accumulation** and reduction.
- DSC: **distributed synchronization controller** with token-based arbitration.
- **Locality-aware protocol**: minimize global sync; favor *cluster-local* updates.

At 2048 DPUs

- **2.9**$\times$ speedup vs. SOTA
- **65%** lower energy
- **92%** parallel efficiency

System Overview



AGAU: Atomic Gradient Accumulation Units

- **256-bit SIMD** for parallel accumulation.
- **Hardware reduction tree**: $O(\log n)$ averaging within a cluster.
- **Local parameter cache** + coherence directory.
- **Atomic primitives**: `FETCH_ADD`, `COMPARE_SWAP`.

Impact

- Cuts intra-cluster communication and software overhead.
- Delivers $\sim$**45%** of end-to-end speedup (see later breakdown).

DSC: Distributed Synchronization Controller

- **Token-based** permissions for cross-cluster updates.
- **Adaptive sync interval** using gradient staleness metrics.
- **Hardware barriers**: local/global modes.
- **Priority arbitration** to avoid hotspots.

Impact

- Reduces global coordination cost;
- $\sim$**35%** of total speedup contribution.

Locality-Aware Communication Protocol

$$T_{\text{comm, DIRECT}} = \begin{cases} L_{\text{local}} + \dfrac{M}{B_{\text{local}}} & \text{(same cluster)} \\ L_{\text{global}} + \dfrac{M}{B_{\text{global}}} + O_{\text{dsc}} & \text{(inter-cluster)} \end{cases}$$

- Prefer **local updates**; escalate selectively when staleness exceeds threshold.
- Hierarchical crossbar minimizes hops; DSC trims sync overhead.

Training Algorithm on DIRECT

Algorithm 1 DIRECT Locality-Aware Training (per epoch)

Require: Learning rate η, batch size B, staleness threshold τ

for each DPU cluster G **in parallel do**
  $g_{\text{local}} \leftarrow \text{AGAU.REDUCE}\left(\sum_{i \in G} \text{COMPUTEGRADIENT}_i(B)\right)$
  if DSC.STALENESS() $> \tau$ **then**
    $t \leftarrow$ DSC.ACQUIRETOKEN()
    $g_{\text{global}} \leftarrow \text{CROSSBARREDUCE}(g_{\text{local}})$
    $w \leftarrow w - \eta \cdot g_{\text{global}}$
    DSC.BROADCASTUPDATE(w); DSC.RELEASETOKEN(t)
  else
    $w \leftarrow w - \eta \cdot g_{\text{local}}$
  end if
end for

Simulation Setup

- **Simulator**: gem5 with PIM extensions (AGAU, DSC, hierarchical crossbar).
- **Scale**: up to **2048** DPUs; cycle-accurate.
- **Baselines**: PIM-Opt (State-of-the-art PIM architecture for accelerating ML workloads); NVIDIA A100 GPU.

Workloads

- **Linear Model**: YFCC100M-HNfc6 (sparse, memory-bound).
- **ResNet-18**: ImageNet (compute-heavy).

Metrics

- Time, Energy, Staleness, Accuracy
- Parallel efficiency @ scale
- Area/Power overheads

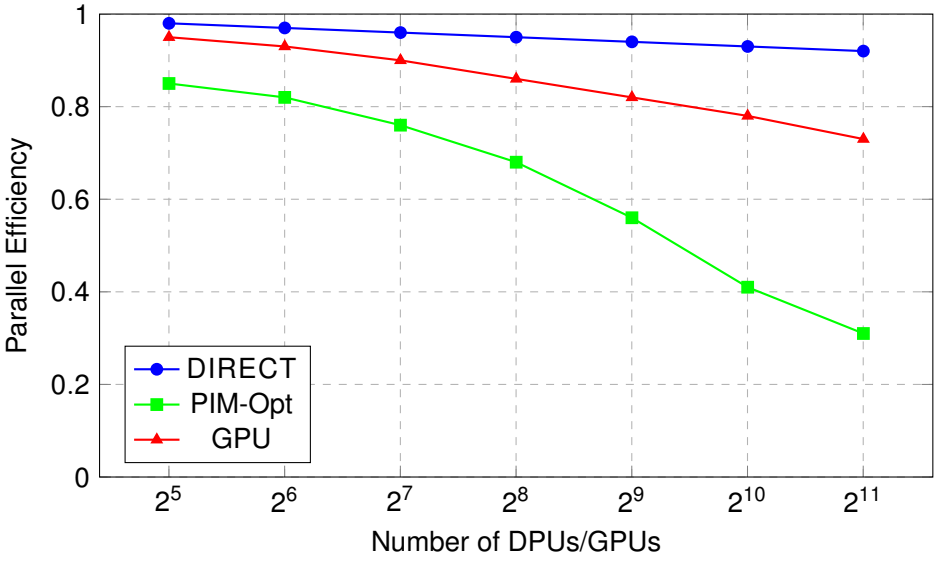
Experimental Configuration

Parameter	Value
DPU Count	Up to 2048
Crossbar Levels	4
Sync Mechanism	Token-based DSC
AGAU Ops	Parallel accumulation & reduction
Granularity	Cycle-accurate
Workloads	ResNet-18, YFCC100M-HNfc6

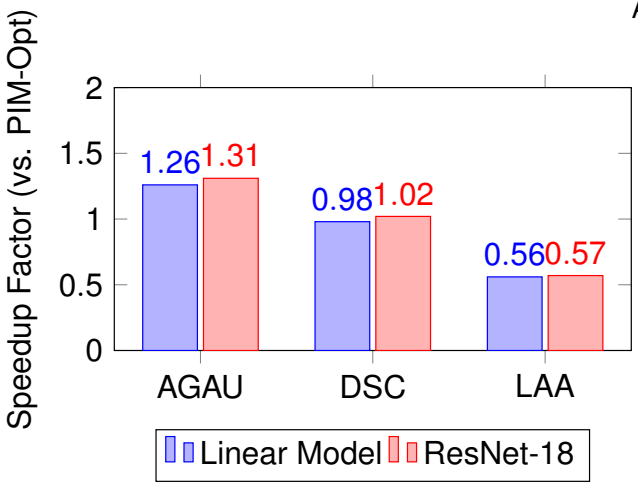
Main Results (Per-Epoch)

Metric	YFCC100M-HNfc6			ResNet-18		
	PIM-Opt	**DIRECT**	GPU	PIM-Opt	**DIRECT**	GPU
Time (s/epoch)	245	**87.5**	120	1240	**428**	650
Energy (J/epoch)	892	**312**	560	4350	**1522**	2740
Staleness (ms)	475	**128**	220	685	**178**	380
Accuracy (%)	84.2	**84.5**	84.3	76.1	**76.3**	76.2
Parallel Efficiency @ 2048						
Compute	0.82	**0.95**	0.88	0.78	**0.93**	0.85
Comm.	0.25	**0.92**	0.72	0.22	**0.89**	0.68
Overall	0.31	**0.93**	0.76	0.28	**0.91**	0.73
Area Overhead (mm^2)	0.42	**0.51**	–	0.42	**0.51**	–
Power Overhead (mW)	185	**205**	–	185	**205**	–

Scaling: Parallel Efficiency vs. System Size



Where the Speedup Comes From



As a percentage of total gains:

- **AGAU**: efficient local reductions ($\sim$ 45%).
- **DSC**: low-overhead global synchronization ($\sim$ 35%).
- **LAA**: locality-aware algorithm ($\sim$ 20%).

Takeaways & Industry Relevance

Key Takeaways

- DIRECT delivers **CPU-free** DPU-to-DPU communication at scale.
- Achieves **2.9×** **speedup**, **65% energy reduction**, **92% efficiency @ 2048** DPUs.
- Minimal overhead: **0.51 mm^2**, **205 mW** (28nm process).

Why it Matters for Industry

- **Cut training costs** by slashing interconnect traffic.
- **Boost sustainability** through large-scale energy savings.
- **Drop-in scalability** for next-gen heterogeneous accelerators.
- **Future-ready** for on-memory AI training workloads.

Thank you! Questions welcome.
