ULTRA ACCELERATOR LINK™

Scale-up Fundamentals

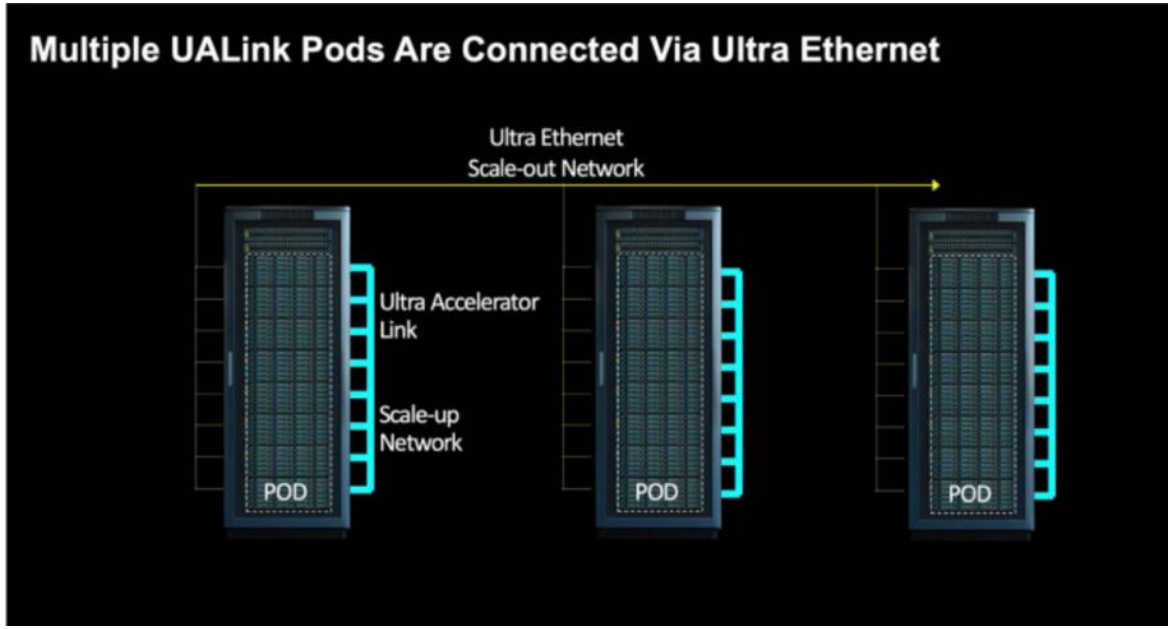
Sharada Yeluri, AVP of Engineering, Astera Labs

Scale-up Fundamentals

- Purpose-built connectivity for XPUs

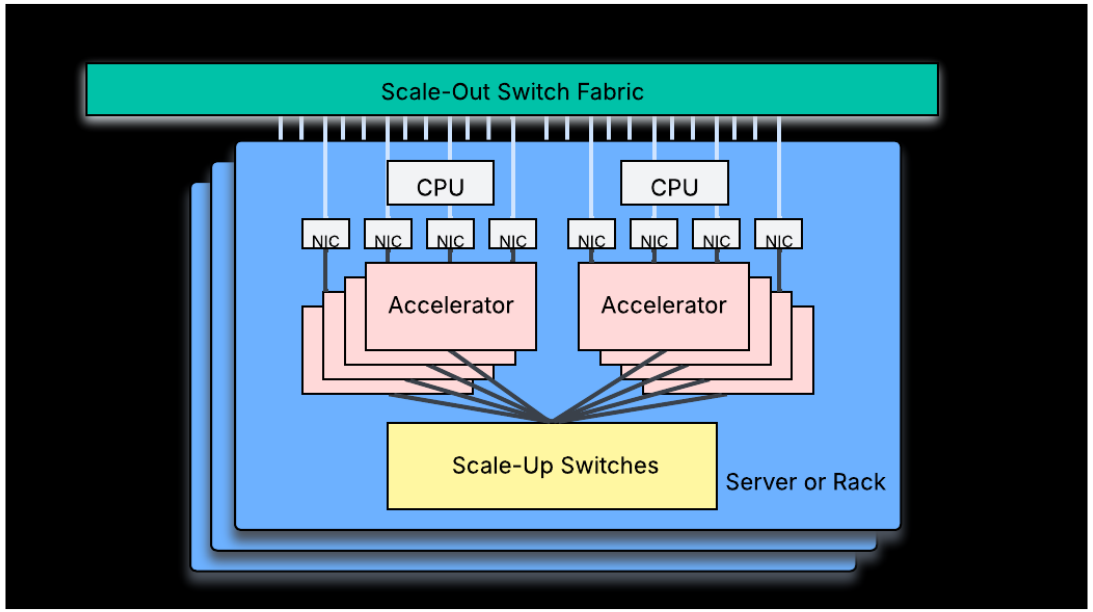
- Rack-level scale
  - Connects XPUs within a single rack or server

- Higher bandwidth
  - 6–12x more than scale-out networks
- Needed for high bandwidth collective communication



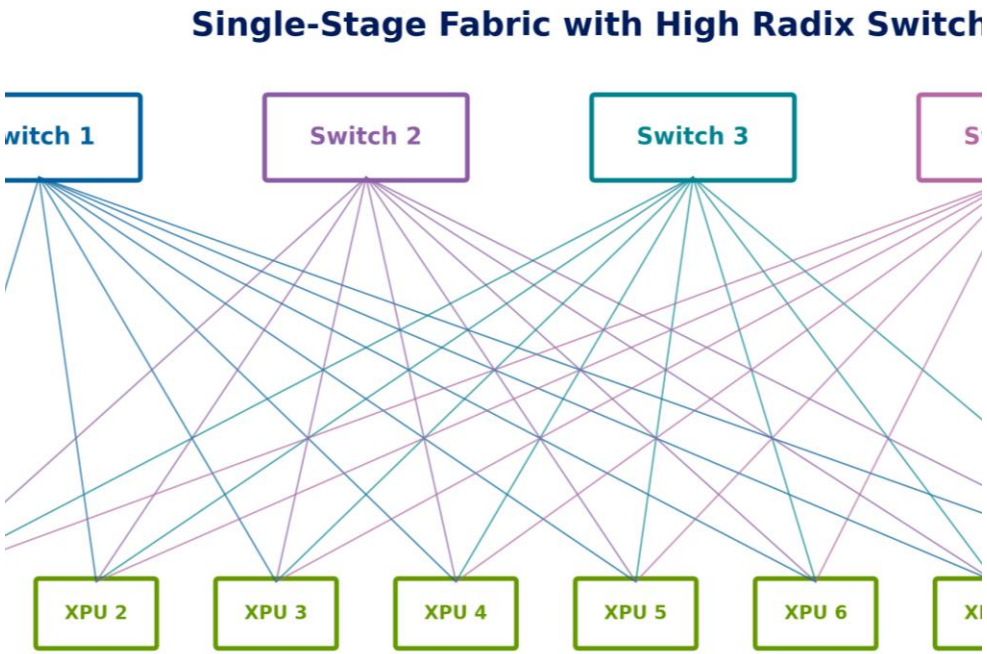
Scale-up Fundamentals

- Collective Communication
  - XPUs exchange partial results
  - Tight coupling - limited opportunities to hide communication latencies
- Load/Store (LS) Semantics
  - No SW overhead
  - Aggregates memory across all XPUs
  - LS identical for remote or local memory
- **Low Latency**
  - Longer latency -> XPU threads stalls
  - For inference, XPU stalls have multiplicative effect for reasoning models



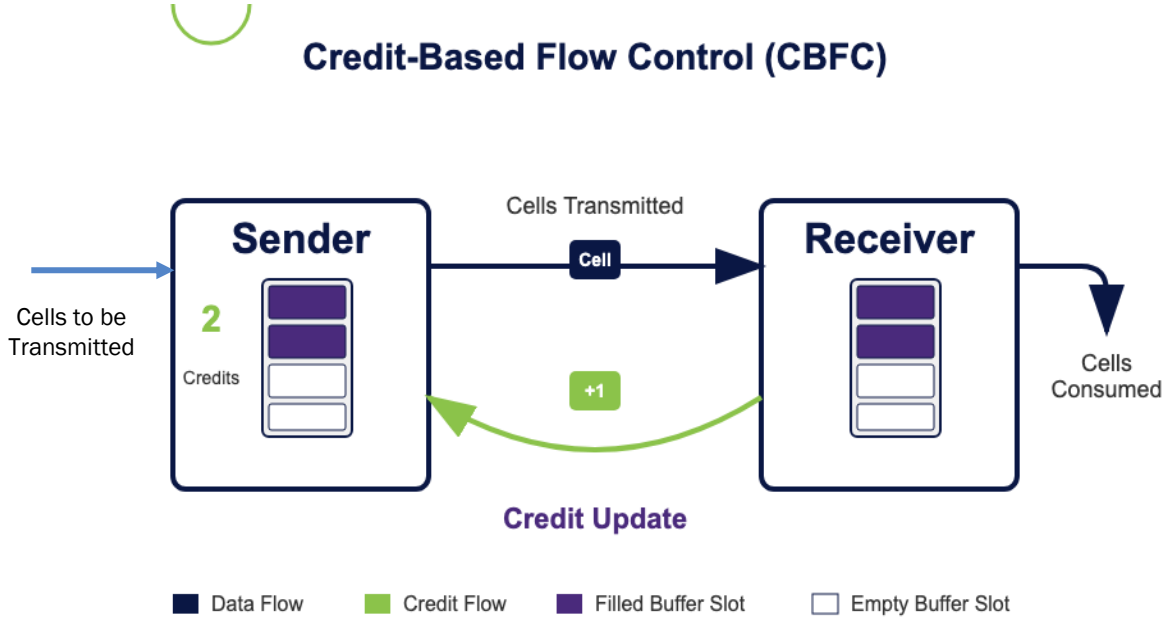
Scale-up Fundamentals

- **Low Jitter**: Compiler scheduling relies on stable and deterministic communication latencies. Low jitter is important
- **Larger radix** enables single-stage fabric – lower latency
- Multiple planes for high bandwidth from each GPU



Scale-up Fundamentals

- Fabric packet drops are fatal to the XPU process threads
- Lossless fabric with hop-by-hop credit-based flow control (link as well as application level)
- Reliable links with FEC and link-level retries are a must
- Non-blocking fabric to eliminate deadlocks: use different VCs/traffic classes for requests and responses



Fabric Requirements

Requirement	Purpose
Ultra-Low Latency & Jitter	Minimize stalls; predictable communication
Non-Blocking	Separate traffic classes avoid deadlocks
High Radix	Many ports enable single-stage fabrics
Lossless Operation	Reliable links with retries and credits
Zero SW Overhead	No RDMA setup for queue pairs and memory registration Direct memory semantics
Memory Ordering	Preserve consistency; avoid reordering 256B aligned memory must not be re-ordered
Bandwidth Efficiency	Maximize payload bits per frame. Reduce protocol and request/ack overhead

UALink - Layers

ULTRA ACCELERATOR LINK™

Layered Design for Efficiency

A multi-layered protocol stack, similar to networking models - Allows for optimization and flexibility at each level of communication.

- **4. Protocol Layer (UPLI)**
  - The top-level interface where the accelerator core connects.
  - It defines the fundamental commands: read, write, and atomic memory operations.
- **3. Transaction Layer (TL)**
  - Segmentation/reassembly of protocol layer transactions to 64B flits
  - Uses features like address compression to maximize bandwidth efficiency.
- **2. Data Link Layer (DL)**
  - It packs data from the layer above into larger, structured "flits" for efficient transfer
  - Supports reliable header with link level retry
- **1. Physical Layer (PHY)**
  - Ethernet Serdes/PHY. It's based on the **IEEE P802.3dj** standard, leveraging proven Ethernet technology for high-speed, reliable signaling (200 GT/s per lane).

Comparison of Open Ecosystems

ULTRA ACCELERATOR LINK™

Attribute	UALink	Ethernet
Scale (single-stage fabric)	1024	1024 or more
PHY	Ethernet	Ethernet
Data Link Layer Frame Sizes	Fixed – 640B	Variable – Up to Ethernet MTU
Link Level Retry (LLR)	Yes	Yes (UEC)
CBFC	Yes, per VC	Yes (UEC), per traffic class
Request/response isolation	yes	Traffic classes used
Ordering	Strict or relaxed ordering. Relaxed for lower latencies	Relaxed ordering is harder to implement

Comparison of Open Ecosystems

ULTRA ACCELERATOR LINK™

Attribute	UALink 200	Ethernet
Unit of switching	64B-256B	Variable packet sizes up to MTU
Bandwidth efficiencies	> 90%	Typically, 80% or less Latency vs. bandwidth trade-off
End-to-End latencies	Superior (less than 1us)	Worse (50% more with standard Ethernet switches)
Jitter	Minimal	High – variable packet sizes
Switch Latencies	Lower due fixed cell size	higher
Switch Power	20-30% less	Higher power
Implementation Complexities	Many choices for UALink controller Ips. Easier to implement the switch fabric with 64B-256B flits	Can re-use modules from standard switches – but may not be optimized for latencies
In Network Compute Support	Easier to implement with 256B flits	Needs unpacking of hundreds of flits inside the packet

Comparison of Open Ecosystems

Attribute	UALink 200	Ethernet
Protocol	UPLI protocol	Any XPU application-level protocol can be wrapped inside the ethernet header.
Security	Supports end-to-end encryption on each req/res channels for confidential computing. Security is built natively	

Call to Action

- There is a need for open/standard Accelerator interface protocols that support load/store semantics with security built in natively.
  - UPLI is the one!
- Carrying UPLI transactions natively with UALink transport/Data-Link is more efficient
- UALink is key to a thriving multi-vendor AI ecosystem
- Choose UALink for your next accelerator interface
- Get involved to deliver UALink 2.0 standard
- UALink website more details: https://ualinkconsortium.org/

THANK YOU