Why Network Notifications Matter - Enabling Resilient Networking for AI

Jeff Tantsura

Nvidia

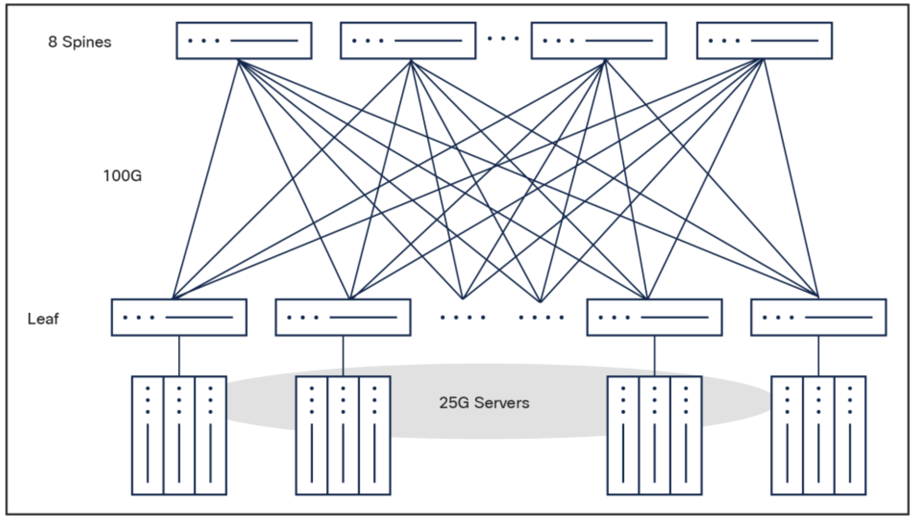
The Growing Demands of AI Workloads

The network connecting GPUs or accelerators during training/inference of large language models.

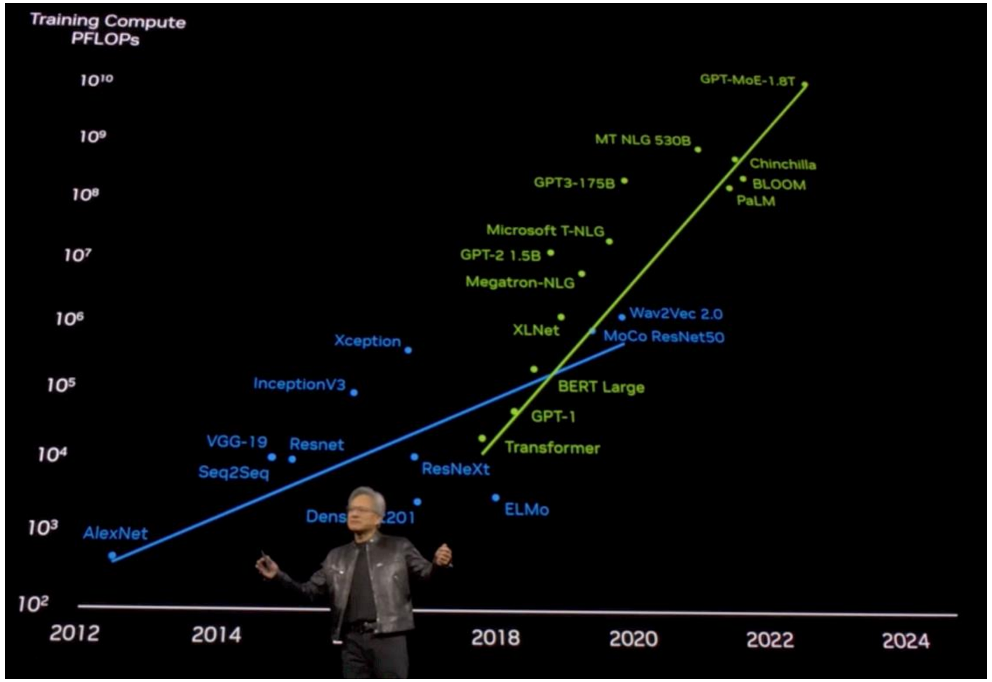
LLM backend fabrics are built for extreme performance at scale — ultra-high throughput, low latency, massive concurrency, and deterministic behavior are all essential.

Model Name	Model Size (Billion Parameters)	Dataset Size (Billion Tokens)	Training ZETA FLOPS (10^21)	GPU	GPU FLOPS/second	GPU utilization	Training Time (in weeks)	Number of GPUs
OPT	175	300	430	H100	3000	0.5	1	474
LLaMA	65	1400	600	H100	3000	0.5	1	662
LLaMA 2	34	2000	400	H100	3000	0.5	1	441
LLaMA2	70	2000	800	H100	3000	0.5	1	882
GPT-3	175	300	420	H100	3000	0.5	1	463
GPT-4* (estimates)	1500	2600	31200	H100	3000	0.5	1	34392

[Demystify Transformers: A Guide to Scaling Laws]



[Data Center Network Fabric]



PFLOPs are growing to catch up with the scaling law of LLMs

Characteristics of AI Training Clusters

- Explosive growth of AI/ML training clusters
- High-performance, low-latency network needs
- Distributed compute across multiple data centers
- Sensitivity to **network failures and congestion**
- Multi-GPU, multi-node distributed training (e.g., Horovod, DeepSpeed)
- Synchronized traffic (e.g., AllReduce)
- Impact of network delays on training convergence and performance
- Need for tight coordination between compute and network layers

Inter-DC Network Realities

- AI workloads often span multiple data centers
- Latency and jitter affect job completion time
- Link failures and congestion are inevitable
- Requires **real-time awareness** of network conditions

The Problem – Gaps in Today's Notification Mechanisms

- Traditional IETF mechanisms (e.g., IGP/BGP convergence) provide routing related information, not much about network quality
- Rely on polling or periodic updates (SNMP, streaming telemetry)
- Lack of near real time, event-driven notifications
- Missed opportunities for real-time response

What Network Notifications Enable

- Real-time congestion signaling
- Application adaptation to network conditions
- Use cases: rerouting traffic, scaling paths, pausing jobs, reducing TX rate, etc.

Types of Network Notifications

- node2node(we are mostly familiar with):
  - L3 – BFD
  - L2 – PFC
  - L2 – ETH OAM (IEEE802.1ag)– not in scoop
- end2end (over the top):
  - RTT probes (not in scoop)
- end2end (network assisted):
  - ECN marking
  - data encoded in INT (set of tech commonly known as in band telemetry)

Types of Network Notifications – work in IETF

- HPCC++ Enhanced High Precision Congestion Control
  - origin: HPCC: High Precision Congestion Control", ACM SIGCOMM19
- IETF
  - draft-miao-ccwg-hpcc – standard track defining the algo
  - draft-miao-ccwg-hpcc-info: informational, inband telemetry encoding formats across all INT encodings (CSIG, IFA, IOAM and INT-P4) and usage

  - Success – has been deployed in production at scale

Example Scenarios

- Loss of a link in a GPU cluster causes training to slow down significantly due to its collective nature
- A node failure may cause complete job failure
- failure could be:
  - fatal (e.g link/node down)
  - temporal (e.g congestion)
- In time failure notifications are a critical enabler for mitigating the impact of failures in AI training clusters.
- The faster the network or system can detect and react to a failure, the smaller is the disruption

Call to Action

- Recognize network notifications as critical infrastructure for AI
- Identify use cases where current protocols fall short
- Support standardization efforts