Hewlett Packard
Enterprise

Will the Cloud Crush HPC? Lessons from Ultra Ethernet

Keith D. Underwood

April 24, 2025

What will become of HPC?

- Much, much larger competition in the room
  - The Cloud
  - AI

- Competing with HPC for:
  - Talent
  - Allocation of sparse resources (GPUs)
  - Area for double-precision floating-point arithmetic on the GPU
  - Silicon design teams to optimize GPUs

- Will HPC simply starve?

Image: AI Generated

Has Anything Changed?

Do you remember when….

- All the money was in mobile?
  - Smartphones
  - Tablets
- Enterprise was the “cash cow”?
  - Really expensive Ethernet NICs with weird features
  - Server CPUs commanded premiums
- PCs emerged as the dominant segment?
  - RAS? ECC? What’s that?
  - I mean… they had a floating-point unit… technically…
- ***Better question: when wasn’t HPC the underdog?***

Image: AI Generated

A New Hope: AI Driving Innovation Needed by HPC

The Ultra Ethernet Story

AI Today: An Opportunity for HPC

What unites us is far greater than what divides us. -JFK

What Unites Us

- Power is a challenge – perhaps ***the*** challenge
- That power turns into heat and answers
  - Mostly heat
- All you get is heat (not answers) if it crashes before you checkpoint
  - Meta estimated a 1.8 hour MTTF for 16K GPU job
- Enough memory capacity at high bandwidth
- Network bandwidth
  - With high network utilization
  - Delivered to the application
  - Without high software overhead
- Network latency and overhead

What Divides Us

- Double precision floating-point
  - Or 32b… or 16b… guessing is close enough, right?
  - But, how much of that do you ***need***?
- Message size
  - Aren't most of your messages at least a megabyte?
  - This is an artifact of the initial focus on training

In early 2022, a group of companies came together with one purpose – to standardize a high-performance network solution for AI & HPC. We called it Ultra Ethernet.

An Overview of Ultra Ethernet

- Challenges encountered with AI at scale motivated the first new transport **standard** in over 20 years
  - Based on semantics and reliability solutions deployed on Ethernet at large scale in HPC
  - Incorporating congestion management solutions developed for Ethernet in the Cloud
  - Initially targeting the needs of AI and HPC “east-west” communications
- The Ultra Ethernet Consortium (UEC) is standardizing:
  - Link layer extensions to provide link level retry and credit-based flow control
  - A new transport – the Ultra Ethernet Transport – encompassing:
    - Semantics oriented around multipath capable, connectionless design
    - Reliability solutions removing the limits on scale while supporting congestion management
    - Encryption technology to enable scalable key state across a million network endpoints
    - Congestion management capabilities exploiting the breadth of paths available in the network
  - Software, Storage, and Management solutions to enable deploying Ultra Ethernet broadly
- The Ultra Ethernet Transport (UET) is designed for the modern datacenter environment
  - Existing Ethernet switches have all the capabilities UET needs

Ultra Ethernet Requirements and Implications

- Support for AI (CCL programming models, SHMEM programming models) and HPC (MPI and SHMEM)
  - CCL programming models have point-to-point, ordered, message-based communication semantics
  - MPI has point-to-point, ordered, tag matched, message-based communication semantics
  - SHMEM uses remote memory access (RMA) semantics with atomics
- Massive scale: 1 million network endpoints
  - Communications state must not increase with the number of endpoints
- Congestion management for standard Ethernet (i.e., without turning on PFC)
  - All proposals leveraged fine grained distribution of packets over available paths
  - Requires native support for out of order packet delivery
- Security from the start: encrypted and authenticated messaging at scale
  - Implies scalable key state
- Enable a wide range of implementation choices
  - Including hardware implementations of reliability and semantic logic
  - Including extremely lightweight accelerator based endpoints
  - Including fully featured high-end endpoints
  - With interoperability amongst all of them

Goals vs Existing Standards: Identifying Gaps

Why did Ultra Ethernet start over?

- Alternative standard network APIs with reliable transports:
  - RC Verbs over RoCE
  - Sockets over TCP (not seriously considered)
- Verbs has user-visible connections for reliable operation with direct data placement
  - Persistent state dedicated per-peer process
  - Scales poorly to 1 million network endpoints
- Specific first-order gaps in the RoCE standard:
  - The RoCE definition was (is) ordered, but congestion management needed to more aggressively multipath through the network leading to unordered packets
    - Re-order buffers are getting expensive as bandwidths increase
  - The congestion management definition for RoCE is not well defined
  - Encryption depends on IPSec, which requires per-peer state
  - Lack of selective ACKs and negative acknowledgements (NACKs)
- Other limitations exist due to the evolution of use cases since the underlying transport was designed almost 30 years ago

Let's make some stone soup

Everyone brought their best ingredients

- From open source:
  - Libfabric as an open, actively evolving network API from HPC
- From AI in the datacenter:
  - Congestion management solutions for Lossy Ethernet
- From HPC:
  - "Connectionless" reliability architecture
  - Semantic layer to support libfabric
- From multiple sources:
  - Nascent definition of scalable shared key security
- Goal: build a modernized transport incorporating the required capabilities from the start – the Ultra Ethernet Transport
  - Out-of-order reliability
  - Congestion management
  - Security
  - Semantics focused on AI and HPC

Image: AI Generated

The Impact of HPC on the Ultra Ethernet Transport

The History of UET Semantics

Libfabric for AI and HPC

- Demonstrated at exascale and in the cloud
  - Shipping on Slingshot
  - Used for AWS EFA
  - Originally created for OPA / OPA2
- Open ecosystem with multiple vendors
- Started as an open-source effort
  - Exposed semantics developed in Portals 4
- Actively being refined for future use cases
  - Added exact tag matching
  - Solidified definition for out-of-order messaging
  - Added AI specific datatypes for atomics

Underpinned by 30 years of HPC development

- Portals was incubated for decades in HPC
  - ASCI Red (Portals 2)
  - CPlant (Portals 3)
  - Cray XT3 (Portals 3.3)
  - Eviden's BXI (Portals 4)
  - Slingshot Cassini NIC (semantics inspired by Portals 4)
- Ultra Ethernet Transport semantics started from Slingshot
  - Added AI specific transactions and datatypes
  - Integrated additional out-of-order messaging
  - Refined field sizes for broader usage

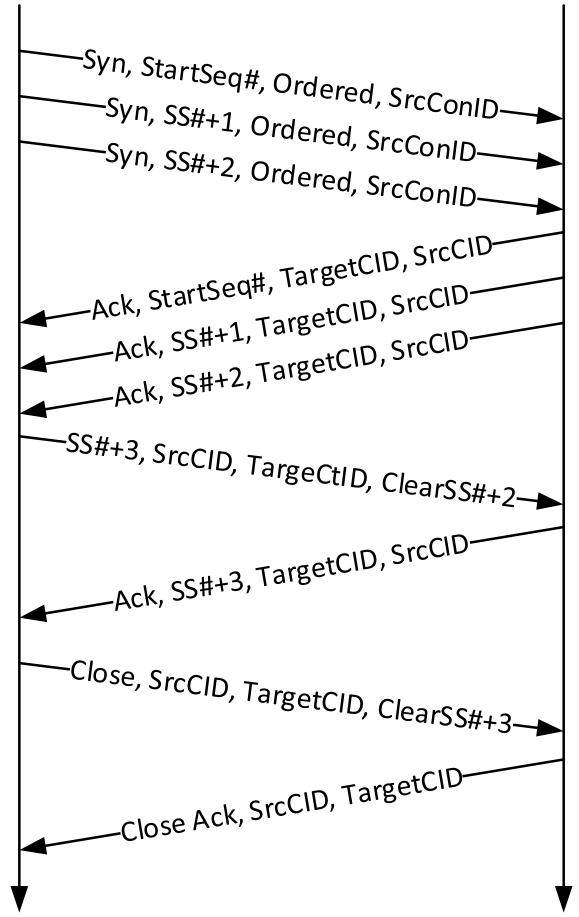
The Impact of HPC on the Ultra Ethernet Transport

Connectionless Reliability Built on Ephemeral State

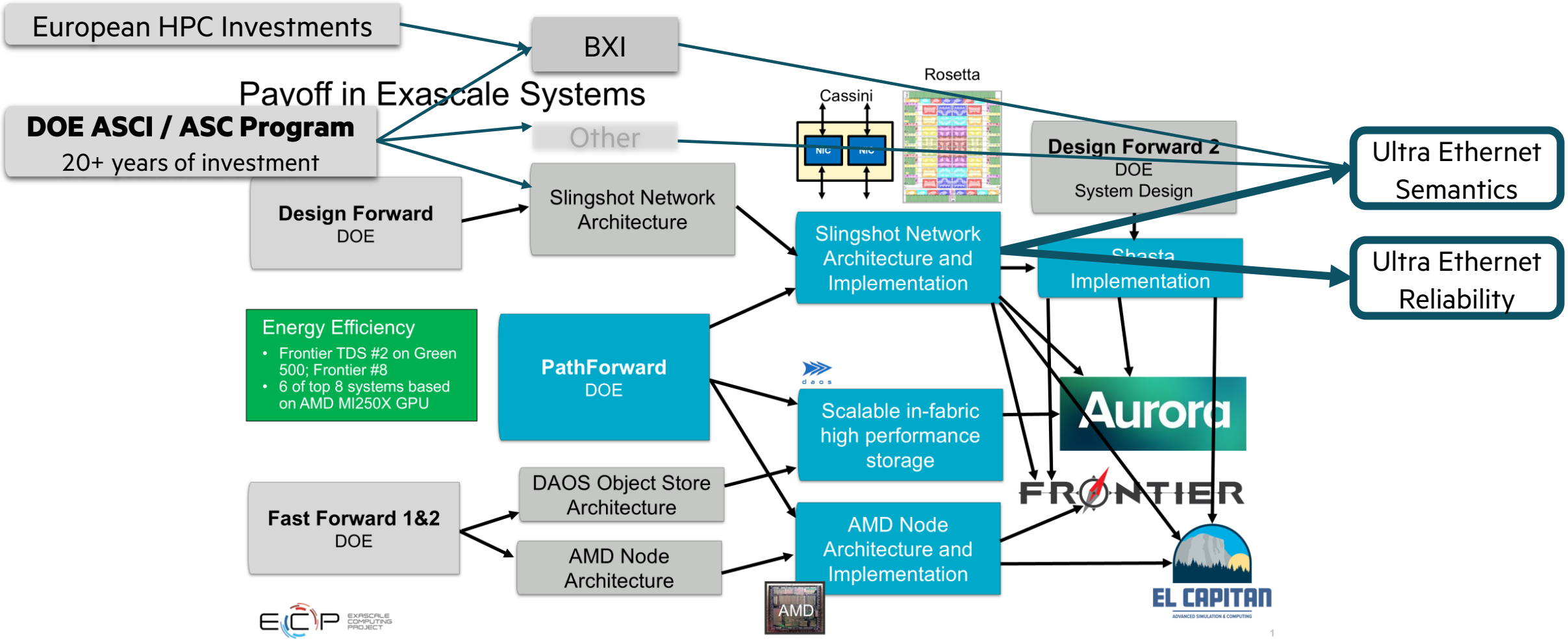
Reliability Using Packet Delivery Contexts (PDCs)

- Connection starts when there is data ready to send
  - Packets are marked with a SYN bit until the first response is received
  - Response contains a target connection ID to use in subsequent packets
  - This is a “0 RTT Startup”
- All acknowledgements must be received before a connection can be closed
  - This means that a Close can clear all state at the target
  - Close operations are acknowledged and will be retried if the acknowledgement is lost
- Targets can request an initiator close a connection
  - For example, if the target connections are overloaded
- Works for ordered and unordered use cases
- Where did this come from?
  - The Slingshot Cassini architecture supported by ECP

PDCs – Illustrated



Decades of HPC Investments Enabled Ultra Ethernet



ECP Slide: Courtesy Bronis R de Supinski

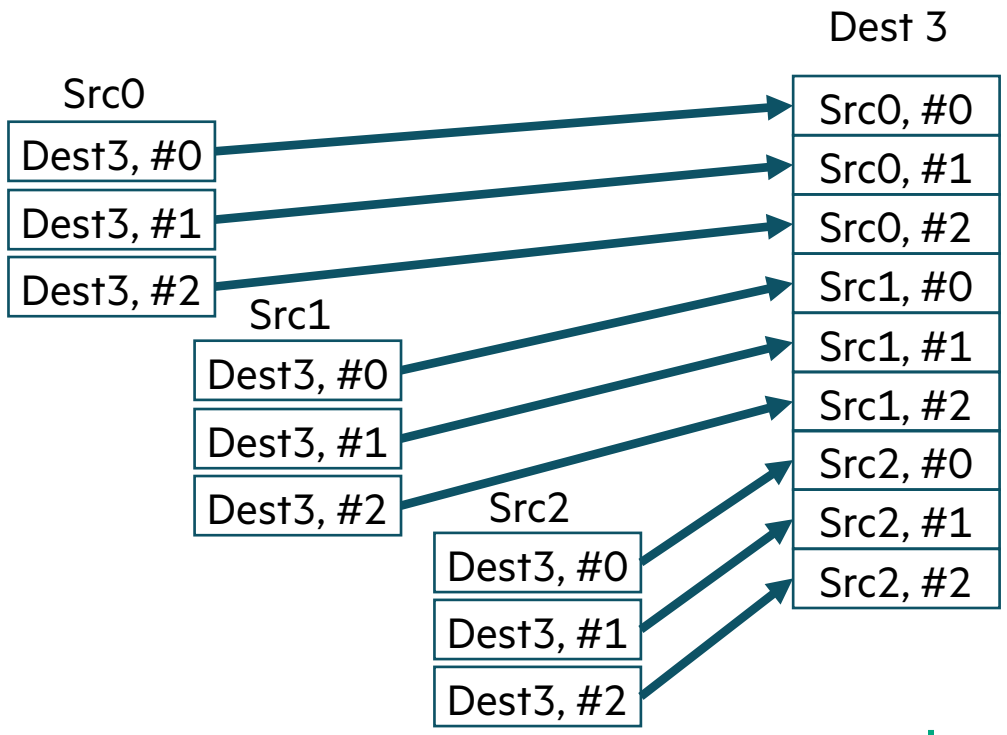
Tag Matching: An HPC Solution Enabling AI Networks

Ultra Ethernet Tag Matching Support

- The Ultra Ethernet Transport supports tag matching
  - Enables offload of tag matching
  - Supports a 64 bit field on the wire (some may be sequestered by libfabric)
  - Also supports a 32 bit initiator field for matching (i.e., to hold a rank)
- Two forms of tag matching supported
  - Exact match: match bits and initiator must match exactly
  - Wildcard match: initiator and match bits may be wildcarded
- Why two forms?
  - Exact match is enough for AI use cases
  - Wildcard matching is needed for MPI support

Tag Matching for AI – Solving Network Ordering

- Uses tag matching to encode required sequencing
  - Uses exact matching to simplify the hardware
- Uses directed receives to make that per source



HPC Does Face Existential Threats

But they are (mostly) not AI…

How will we feed HPC?

If you thought your last electric bill was expensive…

- Do you have a spare decommissioned nuclear site?
- Can we reactivate Bull Run?
- What will 800 MW cost you?

Image: TVA under Creative Commons license

Image: AI Generated

Amazon, Google and Microsoft All Go Nuclear to Power A.I. Ambitions: What to Know

Big Tech is increasingly turning its focus to nuclear energy projects as it seeks out emission-free sources of power.

By Alexandra Tremayne-Pengelly · 10/16/24 8:00am

Source: Observer

https://observer.com/2024/10/amazon-google-microsoft-nuclear-power-ai-ambitions/

Three Mile Island nuclear power plant to return as Microsoft signs 20-year, 835MW AI data center PPA

Site expected to return in 2028, in huge nuclear deal

September 20, 2024 By: Sebastian Moss Have your say

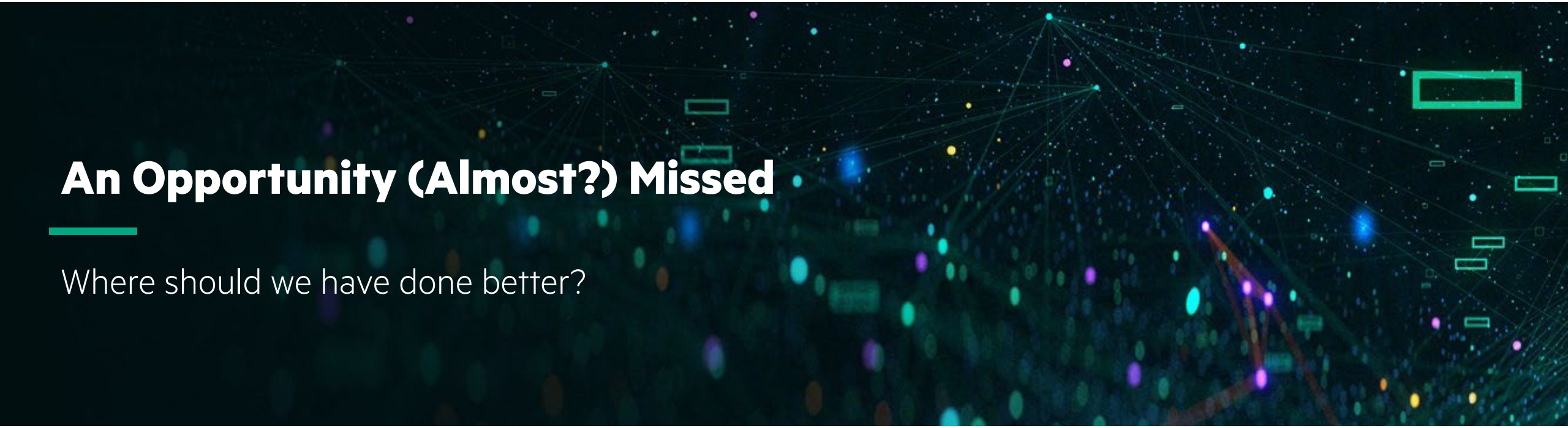
Source: Data Center Dynamics

https://www.datacenterdynamics.com/en/news/three-mile-island-nuclear-power-plant-to-return-as-microsoft-signs-20-year-835mw-ai-data-center-ppa/

Where will we shelter HPC?

The poor fellow can’t stand the heat…

- What will you do with 100MW of heat?
  - In the summer?
  - What if it’s 800 MW?



An Opportunity (Almost?) Missed

Where should we have done better?

Why is NCCL the communication API?

Why isn't it MPI?

NCCL vs MPI

```
ncclResult_t ncclSend(
        const void* sendbuff,
        size_t count,
        ncclDataType_t datatype,
        int peer,
        ncclComm_t comm,
        cudaStream_t stream)

int MPI_Send(
        const void *buf,
        int count,
        MPI_Datatype datatype,
        int dest,
        int tag,
        MPI_Comm comm)
```

What happened?

- MPI moved slowly
  - Nobody who attended the MPI Forum during the MPI-3 effort would have described it as "nimble"
- MPI explicitly rejected ideas that were NCCL-like
  - MPI "subsets" were proposed and rejected
- We failed to see far enough (which didn't need to be very far)
  - Integrating MPI with the GPU programming model should have been "obvious"

What does the future hold?

Prognostications and Suggestions

Will UEC solve all of your network problems?

There is work left to do

AI and HPC? Or AI versus HPC?

- Lossy vs Lossless
  - Clouds really prefer lossy, and AI training uses big flows that can tolerate it
  - HPC prefers lossless, because smaller flows are sensitive to packet loss
- Big messages vs much communication diversity
  - AI training uses a lot of really big messages
  - HPC has done more strong scaling lately
- Exact matching vs MPI matching semantics
  - AI uses CCLs that do not use wildcards for matching messages
  - MPI defines a generic, wildcarded match semantic with strong ordering guarantees

Ultra Ethernet – In Practice

- Profiles help us co-exist and interoperate
  - AI Base
  - AI Full
  - HPC
- Sometimes we built a framework for both
  - Exact matching vs MPI matching was “easy”
- Sometimes we had to defer to the next generation
  - Lossless congestion management
- There will be a ***lot*** of UEC NICs for AI
  - Fewer will focus on HPC

The future of HPC will be…

…a lot like the past

- HPC is still a “big enough” customer for computing elements
  - You will be able to buy something useful (someone will happily take your money)
- You will have to make difficult trade-offs
  - Do you want to do HPC or AI on that compute element?
    - Or AI for Science?
  - How much double-precision is enough?
    - There are two ways to improve the ratio between flops and memory bandwidth
  - How much did you leave for AI for Science?
  - Can you adapt your code to use the AI subset of network semantics?

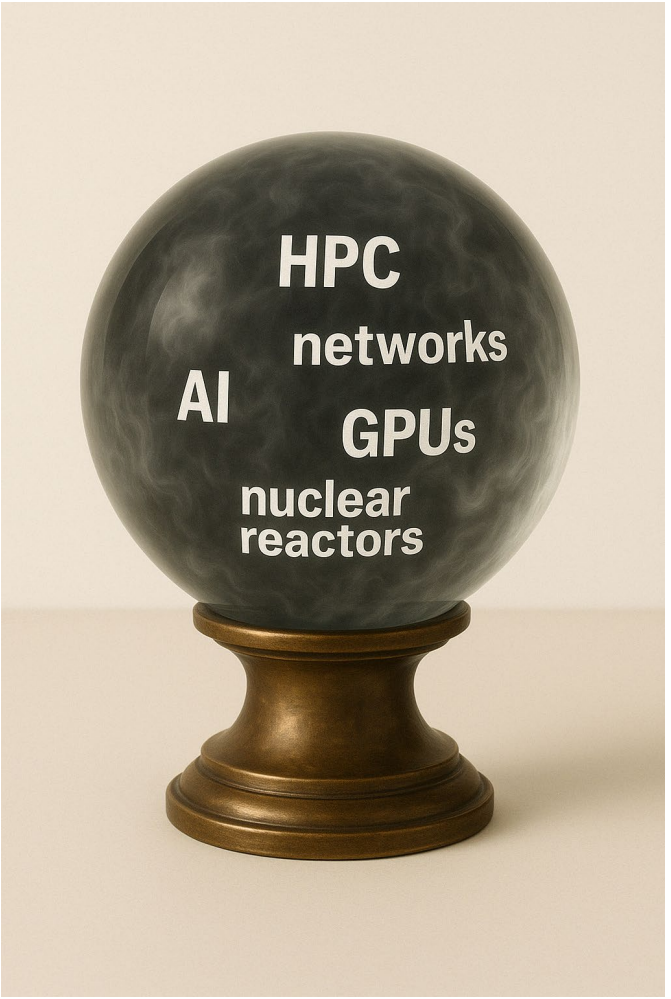


Image: AI Generated

The future of HPC will require…

…focused, strategic investments

- Dan was right: a market opportunity has to be pretty big to entice a company to invest in a new business
  - Standard calculation is “Return on Investment”: ratio of return to the cost
  - Gross margins on future sales have to pay for current spend
  - Discounts future value of money
  - Has to yield a better ROI than other uses of that money
- However: if you offset the “I” with focused NRE investment, it can have a huge impact on ROI
- Where can HPC invest to have impact?
  - HPC still has unique network needs, and the cost of silicon development is going up
  - HPC still needs its own software stack
    - Many cloud providers run their own, proprietary software

Can HPC get what it needs?

Image: AI Generated

The future of HPC will be…

…whatever you make it

The HPC community must identify where it can lead…

- Ultra Ethernet demonstrates the opportunity for AI and HPC to work together
  - And the challenges in doing it
  - Our best path forward is to solve problems together

- Where might we lead?
  - Help AI move past the “age of dense matrices”
  - Make MPI adequate and ubiquitous again
  - Keep SHMEM standard
  - Move beyond implementing every code 3 times
  - AI beyond LLMs
  - Cooperate and deliver elasticity to HPC

…or the view never changes

Image: AI Generated

Thank you