UnifabriX
Leading CXL Smart Memory Technology

FMS the Future of Memory and Storage

CXL and Memory Pools: State of the Union

FMS'24: BMKT-102-1: Memory Markets

Ronen Hyatt, CEO and Chief Architect, UnifabriX

August 2024

Bio

- Expert in **system architectures** with over 25 years of experience
- Leading and delivering **silicon** and **system** designs of:

  CPUs, Accelerator SoCs, IPUs/DPUs, **HPC Fabrics, RDMA,**

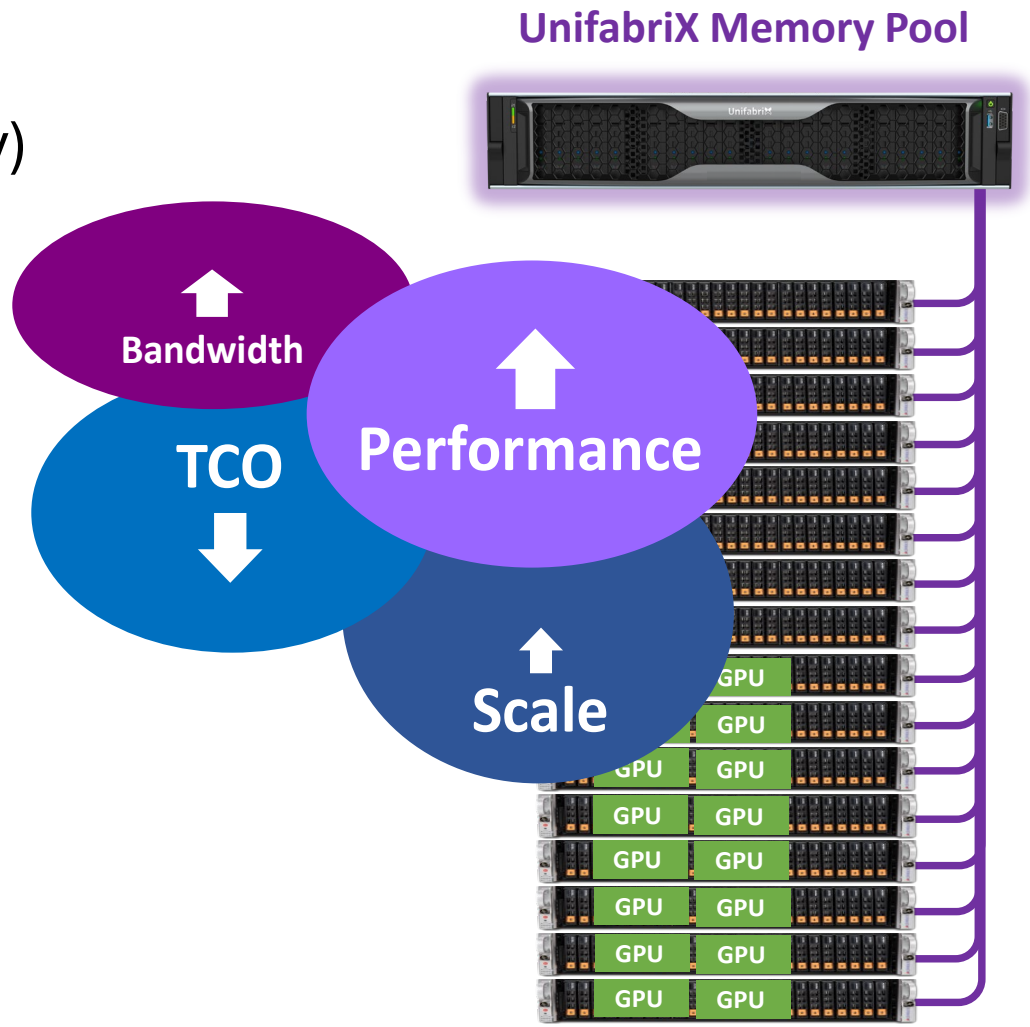
  Programmable Ethernet NIC and Switch, **CXL**
- Held multiple CTO and Lead Architect positions
- **CEO** and **Chief Architect** at **UnifabriX**,

  a system and silicon startup targeting the Memory Wall

  with CXL-based **Software-Defined Memory Pools** and **CXL Fabrics**.
- More than 40 patents (some pending)
- MSc and BSc in Computer Engineering from Technion Institute of Technology

CXL Memory: The Killer Application is Memory Pooling

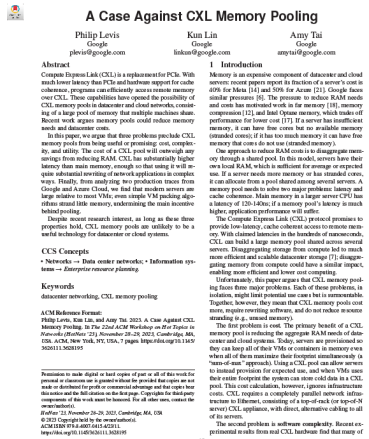
- High-Bandwidth Memory provisioning
- Performance acceleration (+BW) (+Capacity)
- Significant savings in CAPEX and TCO
- Elastic on-demand capacity expansion
- In-Memory Analytics
- Adaptive Sharing



CXL Memory on the Hype Cycle

CXL on the Gartner Hype Cycle?

A Case Against CXL Memory Pooling?



A Case Against CXL Memory Pooling

Philip Levis, Google, plevis@google.com
Kun Lin, Google, linkun@google.com
Amy Tai, Google, amytai@google.com

Abstract

Compute Express Link (CXL) is a replacement for PCIe. With much lower latency than PCIe and hardware support for cache coherence, programs can efficiently access remote memory over CXL. These capabilities have opened the possibility of CXL memory pools in datacenter and cloud networks, consisting of a large pool of memory that multiple machines share. Recent work argues memory pools could reduce memory needs and datacenter costs.

In this paper, we argue that three problems preclude CXL memory pools from being useful or promising: cost, complexity, and utility. The cost of a CXL pool will outweigh any savings from reducing RAM. CXL has substantially higher latency than main memory, enough so that using it will require substantial rewriting of network applications in complex ways. Finally, from analyzing two production traces from Google and Azure Cloud, we find that modern servers are large relative to most VMs; even simple VM packing algorithms strand little memory, undermining the main incentive behind pooling.

Despite recent research interest, as long as these three properties hold, CXL memory pools are unlikely to be a useful technology for datacenter or cloud systems.

CCS Concepts

• Networks → Data center networks; • Information systems → Enterprise resource planning.

Keywords

datacenter networking, CXL, memory pooling

ACM Reference Format:

1 Introduction

Memory is an expensive component of datacenter and cloud servers: recent papers report its fraction of a server's cost is 40% for Meta [14] and 50% for Azure [21]. Google faces similar pressures [8]. The pressure to reduce RAM needs and costs has motivated work in far memory [18], memory compression [12], and Intel Optane memory, which trades off performance for lower cost [17]. If a server has insufficient memory, it can have free cores but no available memory (stranded cores); if it has too much memory it can have free memory that cores do not use (stranded memory).

One approach to reduce RAM costs is to disaggregate memory through a shared pool. In this model, servers have their own local RAM, which is sufficient for average or expected use. If a server needs more memory or has stranded cores, it can allocate from a pool shared among several servers. A memory pool needs to solve two major problems: latency and cache coherence. Main memory in a larger server CPU has a latency of 120-140ns; if a memory pool's latency is much higher, application performance will suffer.

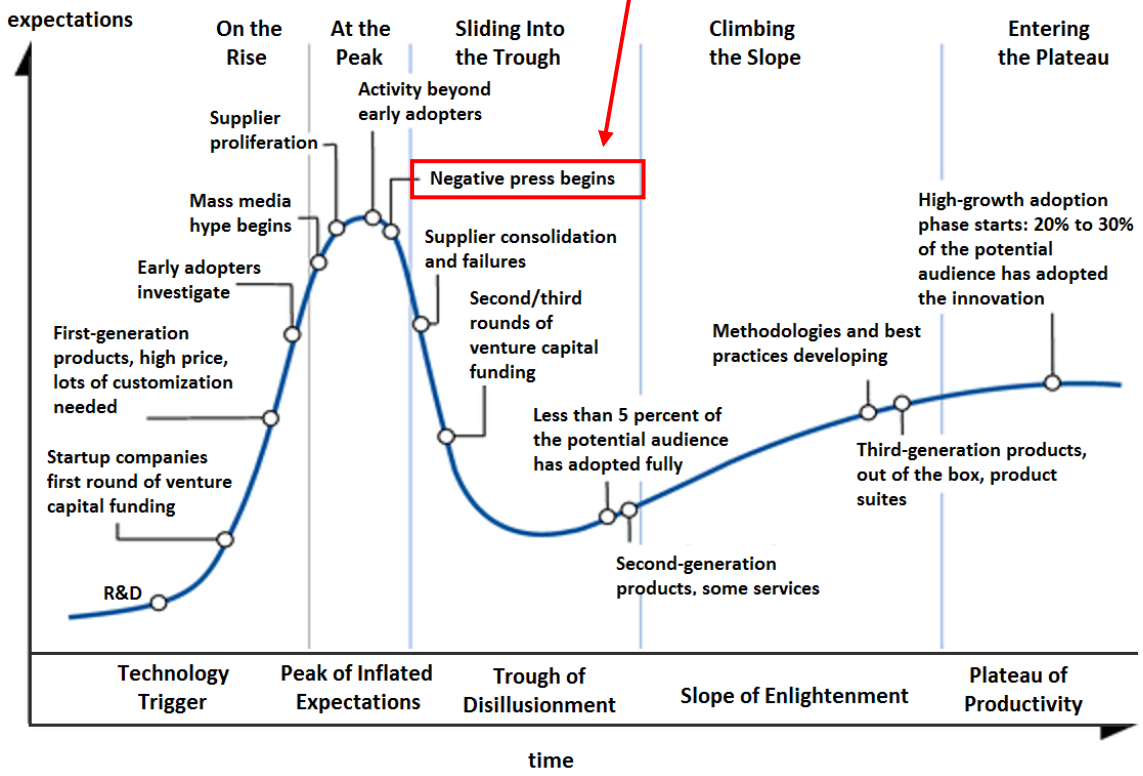
The Compute Express Link (CXL) protocol promises to provide low-latency, cache coherent access to remote memory. With claimed latencies in the hundreds of nanoseconds, CXL can build a large memory pool shared across several servers. Disaggregating storage from compute led to much more efficient and scalable datacenter storage [7]; disaggregating memory from compute could have a similar impact, enabling more efficient and lower cost computing.

Unfortunately, this paper argues that CXL memory pooling faces three major problems. Each of these problems, in isolation, might limit potential use cases but is surmountable. Together, however, they mean that CXL memory pools cost more, require rewriting software, and do not reduce resource stranding (e.g., unused memory).

The first problem is cost. The primary benefit of a CXL memory pool is reducing the aggregate RAM needs of datacenter and cloud systems. Today, servers are provisioned so they can keep all of their VMs or containers in memory even when all of them maximize their footprint simultaneously (a "sum-of-max" approach). Using a CXL pool can allow servers to instead provision for expected use, and when VMs need their entire footprint the system can store cold data in a CXL pool. This cost calculation, however, ignores infrastructure costs. CXL requires a completely parallel network infrastructure to Ethernet, consisting of a top-of-rack (or top-of-N server) CXL appliance, with direct, alternative cabling to all of its servers.

The second problem is software complexity. Recent experimental results from real CXL hardware find that many of

Negative press begins

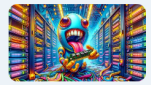


CXL is Dead in the AI Era?

SemiAnalysis
4,848 followers
1mo •

+ Follow

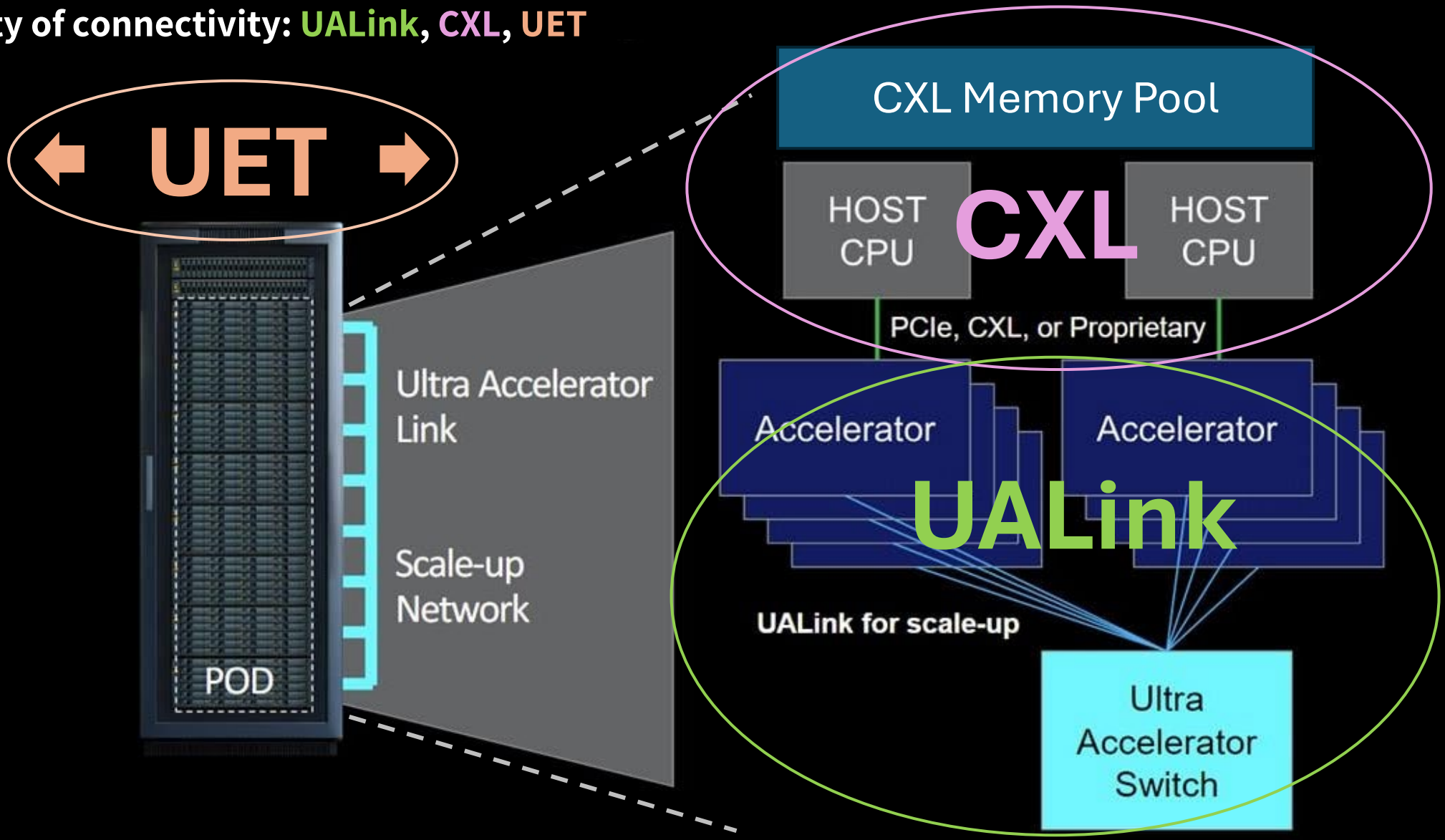
CXL Is Dead In The AI Era - AI Accelerator Beachfront Considerations, Memory Pooling Negatives, Custom Silicon Adoption
Two posts in 1!
Astera Labs IPO - The Next Connectivity Superhero or Steamrolled By Competition? Bottoms Up Model, Units, ASP, Revenue, By Hyperscaler Analysis, EPS & Cashflow, Competitive Analysis
https://lnkd.in/dps8fHU4
https://lnkd.in/dtpJSmnc



CXL Is Dead In The AI Era
semianalysis.com • 1 min read

CXL for AI? Definitely!

The Trinity of connectivity: UALink, CXL, UET



What about CXL Memory Pooling?

Testing the assumptions: Going above and beyond the Abstract

A Case Against CXL Memory Pooling

Philip Levis
Google
plevis@google.com

Kun Lin
Google
linkun@google.com

Amy Tai
Google
amytai@google.com

Abstract

Compute Express Link (CXL) is a replacement for PCIe. With much lower latency than PCIe and hardware support for cache coherence, programs can efficiently access remote memory over CXL. These capabilities have opened the possibility of CXL memory pools in datacenter and cloud networks, consisting of a large pool of memory that multiple machines share. Recent work argues memory pools could reduce memory needs and datacenter costs.

In this paper, we argue that three problems preclude CXL memory pools from being useful or promising: cost, complexity, and utility. The cost of a CXL pool will outweigh any savings from reducing RAM. CXL has substantially higher latency than main memory, enough so that using it will require substantial rewriting of network applications in complex ways. Finally, from analyzing two production traces from Google and Azure Cloud, we find that modern servers are large relative to most VMs; even simple VM packing algorithms strand little memory, undermining the main incentive behind pooling.

Despite recent research interest, as long as these three properties hold, CXL memory pools are unlikely to be a useful technology for datacenter or cloud systems.

CCS Concepts

• **Networks** → **Data center networks**; • **Information systems** → *Enterprise resource planning*.

Keywords

datacenter networking, CXL memory pooling

ACM Reference Format:

Philip Levis, Kun Lin, and Amy Tai. 2023. A Case Against CXL Memory Pooling. In *The 22nd ACM Workshop on Hot Topics in Networks (HotNets '23), November 28–29, 2023, Cambridge, MA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626111.3628195

HotNets '23, November 28–29, 2023, Cambridge, MA, USA

ACM ISBN 979-8-4007-0415-4/23/11.
https://doi.org/10.1145/3626111.3628195

1 Introduction

Memory is an expensive component of datacenter and cloud servers: recent papers report its fraction of a server's cost is 40% for Meta [14] and 50% for Azure [21]. Google faces similar pressures [6]. The pressure to reduce RAM needs and costs has motivated work in far memory [18], memory compression [12], and Intel Optane memory, which trades off performance for lower cost [17]. If a server has insufficient memory, it can have free cores but no available memory (stranded cores); if it has too much memory it can have free memory that cores do not use (stranded memory).

One approach to reduce RAM costs is to disaggregate memory through a shared pool. In this model, servers have their own local RAM, which is sufficient for average or expected use. If a server needs more memory or has stranded cores, it can allocate from a pool shared among several servers. A memory pool needs to solve two major problems: latency and cache coherence. Main memory in a larger server CPU has a latency of 120-140ns; if a memory pool's latency is much higher, application performance will suffer.

The Compute Express Link (CXL) protocol promises to provide low-latency, cache coherent access to remote memory. With claimed latencies in the hundreds of nanoseconds, CXL can build a large memory pool shared across several servers. Disaggregating storage from compute led to much more efficient and scalable datacenter storage [7]; disaggregating memory from compute could have a similar impact, enabling more efficient and lower cost computing.

Unfortunately, this paper argues that CXL memory pooling faces three major problems. Each of these problems, in isolation, might limit potential use cases but is surmountable. Together, however, they mean that CXL memory pools cost more, require rewriting software, and do not reduce resource stranding (e.g., unused memory).

The first problem is **cost**. The primary benefit of a CXL memory pool is reducing the aggregate RAM needs of datacenter and cloud systems. Today, servers are provisioned so they can keep all of their VMs or containers in memory even when all of them maximize their footprint simultaneously (a "sum-of-max" approach). Using a CXL pool can allow servers to instead provision for expected use, and when VMs uses their entire footprint the system can store cold data in a CXL pool. This cost calculation, however, ignores infrastructure costs. CXL requires a completely parallel network infrastructure to Ethernet, consisting of a top-of-rack (or top-of-N server) CXL appliance, with direct, alternative cabling to all of its servers.

The second problem is **software complexity**. Recent experimental results from real CXL hardware find that many of

Page 4: Assumptions regarding DRAM costs

sending responses back to servers. A standard CXL memory device (e.g., a Astera Leo [1] or Intel device [10]) uses 16 lanes. At PCIe Gen5 speeds this is 480Gbps. A 16-server pool therefore processes data at 7.6Tbps.

A modern, low-end, 32-port 200Gbps Ethernet switch such as the Mellanox MSN3700-VS2F0 costs $38,500. [2] DDR5 RAM today is ≈ 3$/GB. For the CXL pool device to break even with its RAM savings, it must save 12.6TB of RAM Assuming Pond's optimistic 9% reduction, to break even with just the switch, the servers must have $\frac{12.6TB}{0.09}$ = 140TB of RAM in aggregate (using Pond would reduce this to 127TB). For a 32-node pool, 127TB, means 4TB per server. A dual-socket AMD Genoa server, the standard next-generation system for cloud providers, has 384 vCPUs. At 4TB/server, there is > 10GB of RAM per Genoa vCPU, more than high-RAM VMs provide. You have to buy considerably more RAM for Pond's RAM savings to pay for themselves: you are better

Bring me some real DRAM to see

Page 4: Assumptions regarding DRAM costs: flat $3/GB across speeds and capacities

sending responses back to servers. A standard CXL memory device (e.g., a Astera Leo [1] or Intel device [10]) uses 16 lanes. At PCIe Gen5 speeds this is 480Gbps. A 16-server pool therefore processes data at 7.6Tbps.

A modern, low-end, 32-port 200Gbps Ethernet switch such as the Mellanox MSN3700-VS2F0 costs $38,500. [2] DDR5 RAM today is ≈ 3$/GB. For the CXL pool device to break even with its RAM savings, it must save 12.6TB of RAM Assuming Pond's optimistic 9% reduction, to break even with just the switch, the servers must have $\frac{12.6TB}{0.09}$ = 140TB of RAM in aggregate (using Pond would reduce this to 127TB). For a 32-node pool, 127TB, means 4TB per server. A dual-socket AMD Genoa server, the standard next-generation system for cloud providers, has 384 vCPUs. At 4TB/server, there is > 10GB of RAM per Genoa vCPU, more than high-RAM VMs provide. You have to buy considerably more RAM for Pond's RAM savings to pay for themselves: you are better

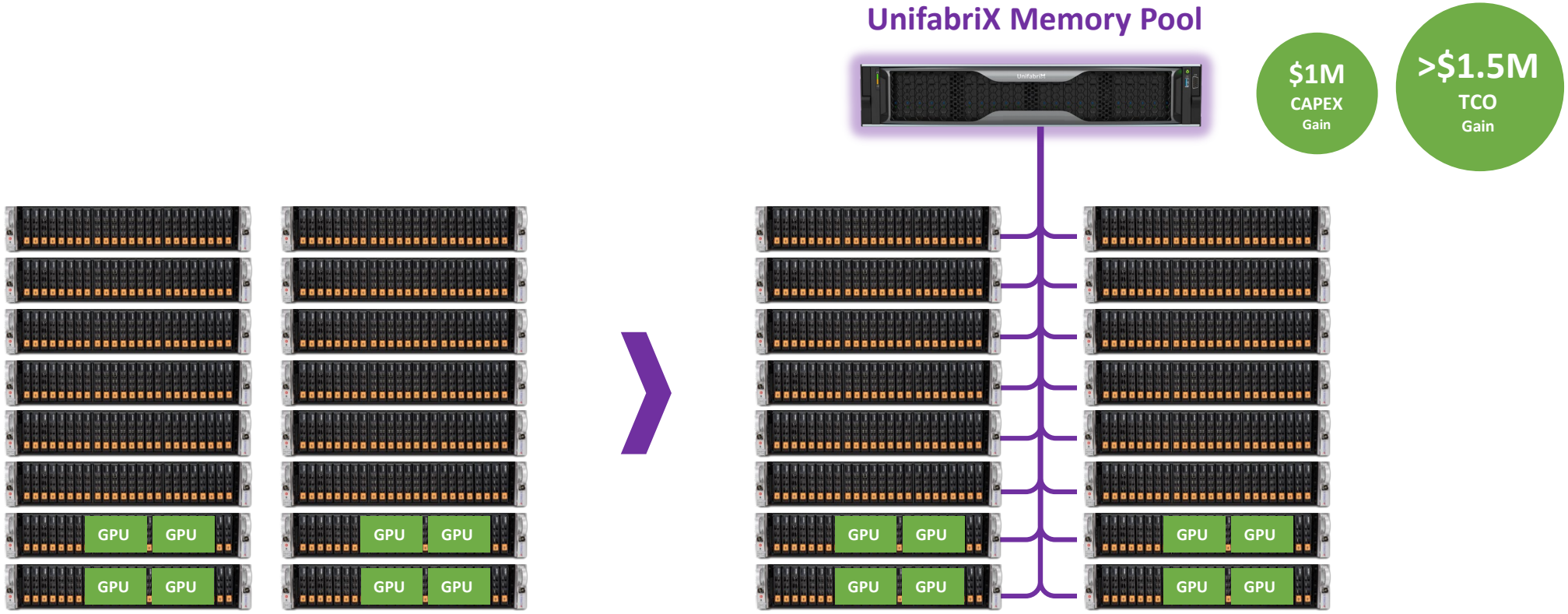
Real-world out there: Houston, we have a problem! We found a curve!

DDR5 Pricing

64GB	4800	$249	$3.89/GB
	5600	$276	$4.31/GB
96GB	4800	$433	$4.51/GB
128GB	4800	$2041	$15.95/GB
256GB	4800	$4265	$16.66/GB

Meanwhile, out there in the Real World : CXL TCO-ware

Use case analysis of "X": $1M savings in CAPEX, >$1.5 savings in TCO



Setup A (Reference)
- 16 x Servers (6TB each)
- Memory Utilization: <30%
- Total Capacity: 96TB
- Total Memory Cost: **$1.6M**

-Overprovisioning
-Extra power
-Memory Stranding
-Rigid allocations

Setup B (with Memory Pool)
- 16 x Servers (2.25TB each)
- Memory Pool (30TB)
- Total Capacity: 66TB
- Total Memory Cost: **$670K**

+Performance boost
+On-demand Memory Bandwidth
+Dynamic infrastructure scaling and agility
+Reduced thermal dissipation

Overview

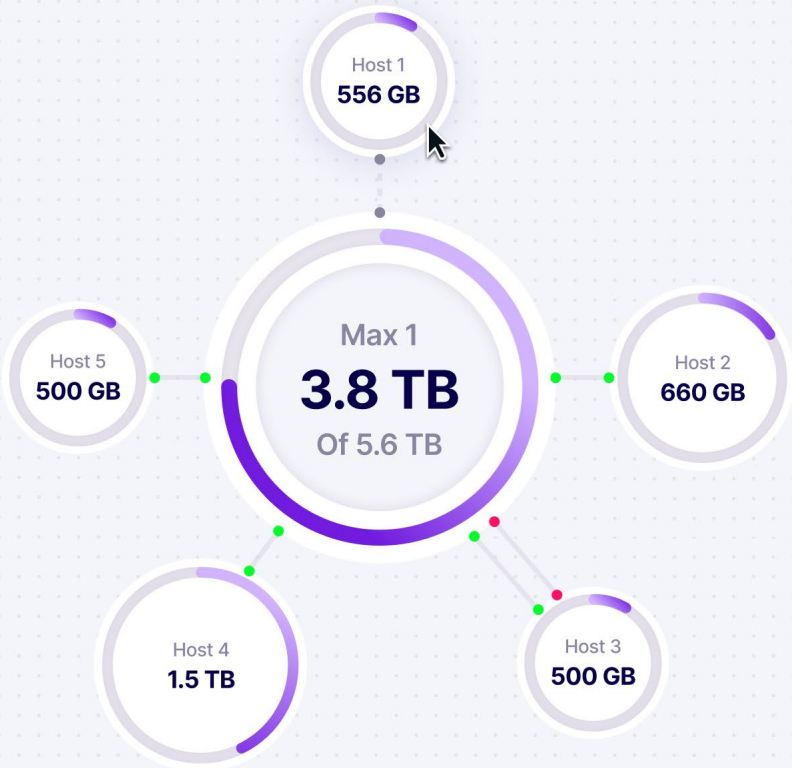
3,892 GB
(Max.01) Total Memory Provisioned

48%
(Max.01) Memory Utilization

5
(Max.01) CXL Ports

5
(Max.01) Active Hosts

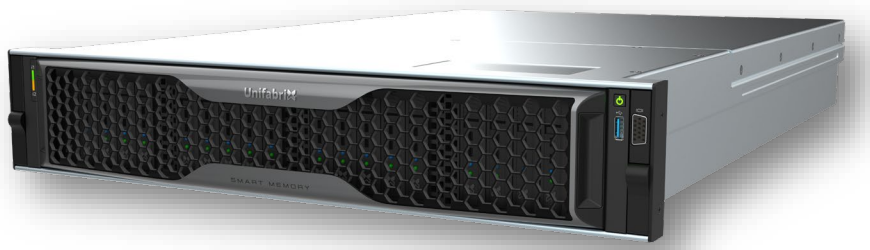
Search..



Optimal · No connection · No link · Degraded

Ronen Hyatt / CXL and Memory Pools: State of the Union
FMS'24: BMKT-102-1: Memory Markets

FMS Unifabrix



Thankyou

UnifabriX