In Computer Architecture, We Don't Change the Questions, We Change the Answers

Mark D. Hill, University of Wisconsin-Madison

Abstract: When I was a new professor in the late 1980s, my senior colleague Jim Goodman told me, "On the computer architecture PhD qualifying exam, we don't change the questions, we only change the answers." More generally, I now augment this to say, *"In computer architecture, we don't change the questions, application and technology innovations change the answers, and it's our job to recognize those changes."* Eternal questions this talk will sample are how best to do the following interacting factors: compute, memory, storage, interconnect/networking, security, power, cooling and one more. The talk will not provide the answers but leave that as an audience exercise.

Biography: Mark D. Hill is the Gene M. Amdahl and John P. Morgridge Professor Emeritus of Computer Sciences at the University of Wisconsin-Madison (http://www.cs.wisc.edu/~markhill), following his 1988-2020 service in Computer Sciences and Electrical and Computer Engineering. His research interests include parallel-computer system design, memory system design, and computer simulation. Hill's work is highly collaborative with over 170 co-authors. He received the 2019 Eckert-Mauchly Award and is a fellow of AAAS, ACM, and IEEE. He served on the Computing Community Consortium (CCC) 2013-21 including as CCC Chair 2018-20, Computing Research Association (CRA) Board of Directors 2018-20, and Wisconsin Computer Sciences Department Chair 2014-2017. Hill was Partner Hardware Architect at Microsoft (2020-2024) where he led software-hardware some pathfinding for Azure. Hill has a PhD in computer science from the University of California, Berkeley.

In Computer Architecture, We Don’t Change the Questions, We Change the Answers

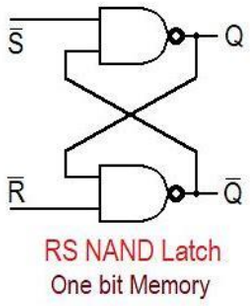
Mark D. Hill
University of Wisconsin-Madison Professor Emeritus

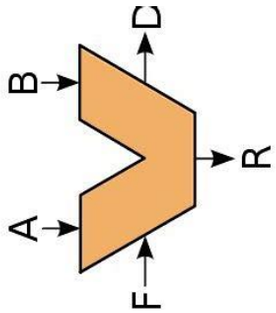
Computer Systems Workshop
IISc, Silicon Valley, India, October 2025

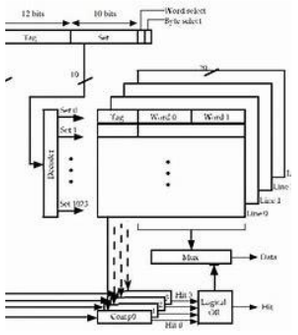
Computer Architecture: Big Picture of Computer Hardware

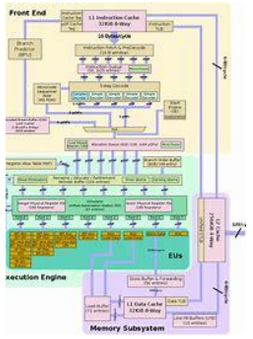
Components → Systems

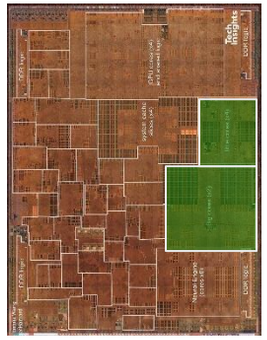
Gates → ALU → Functional Block → Core → SoC → Server → Data Center











Computer Architects: Components → Systems

2020–2024: Hardware-software pathfinding for Azure
Now industrial consultant

A View of Computing's "Stack"

Problem & Algorithms

Applications

DBMSs & Middleware

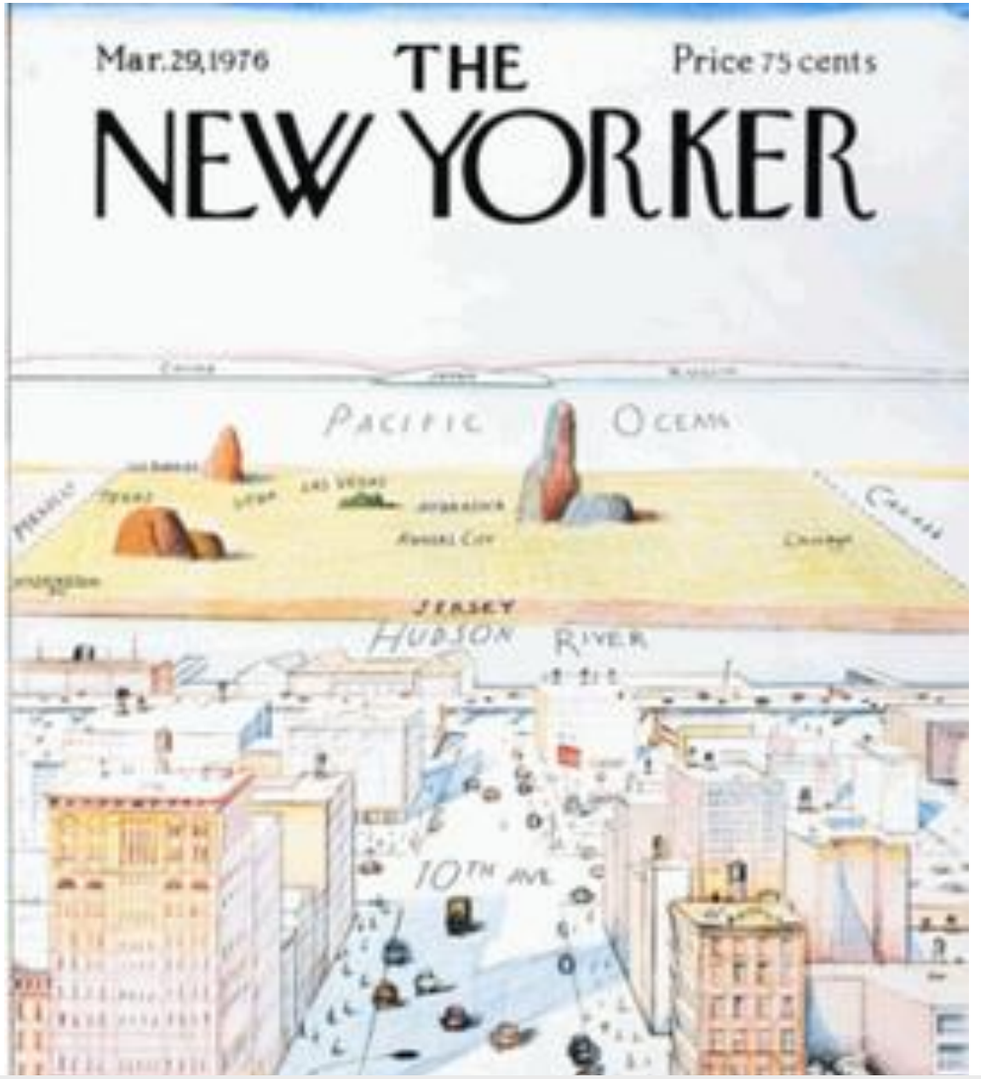
Runtime & Compiler

Operating System

(Micro) Architecture

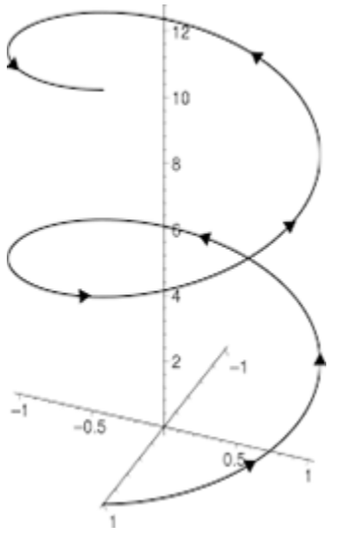
Hardware

Materials & Fabrication



As technology scaling slows, dramatic perf/cost gains needed will require layer experts to work together!

A Commercial Computing Company Helix



IBM digital etc.

Adobe etc.

ORACLE etc.

Microsoft etc.

DELL etc.

intel etc.

Apple

amazon

Google

Meta

etc. Microsoft

Pre-microprocessor Era

Medium tech progress

Users share

Comp layers nascent

Vertical companies

Microprocessor Era

Amazing tech progress

Per-user devices

Comp layers rigid

Horizontal companies

Cloud & Mobile Era

Medium tech progress

Users share cloud, not dev

Cross-layer opt req'd

Vertical companies

New Assistant Professor [1988]

Mark Hill:
How do we update questions for the computer architecture PhD qualifying exam?

Jim Goodman:
We don't change the questions.
We change the answers.

My Current View

In computer architecture,

We don't change the questions

`

Applications & technology innovations change the answers

It's our job to recognize those changes

E.g., Single Instruction Multiple Data (SIMD): 1960s → GP-GPUs

This talk discusses these eternal questions; answers TBD by you!

Computer Architecture's Eternal Questions & Outline

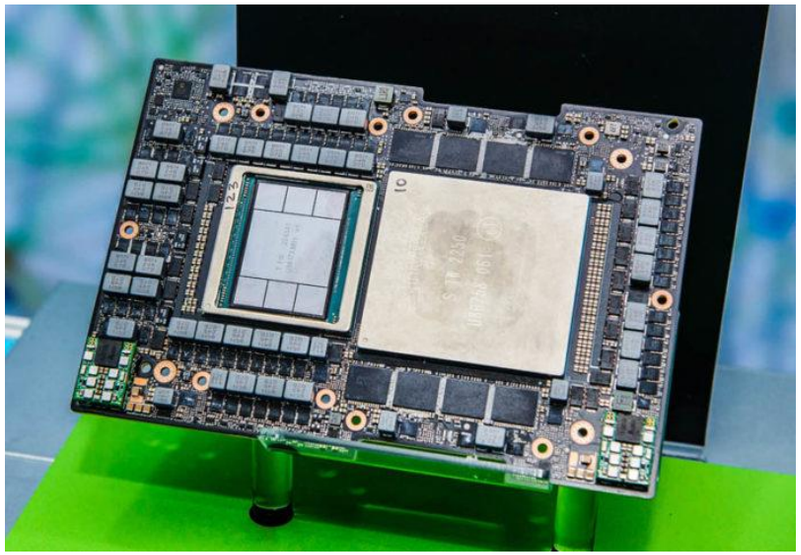
How best to do these interacting factors:

1. **Compute (longest)**
2. **Memory (longer)**
3. Interconnect/networking
4. ~~Storage~~
5. Security
6. Power
7. Cooling
8. *Bonus new question*

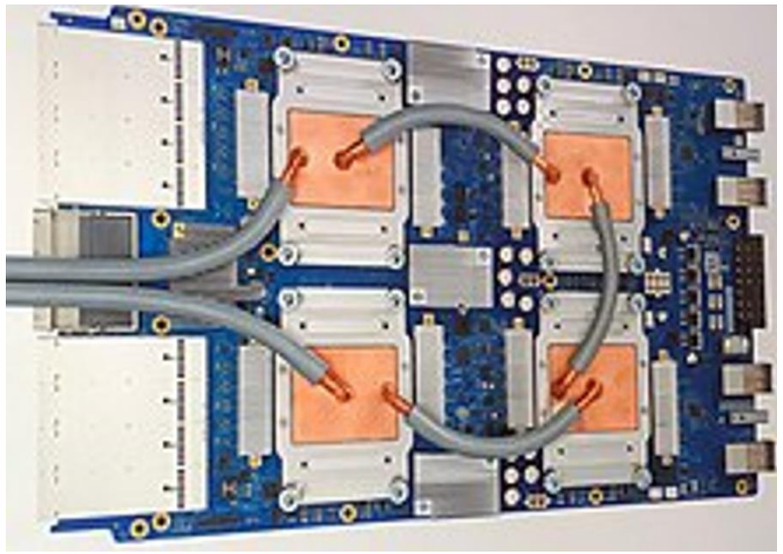
Compute: Accelerators, e.g., Deep Learning

End of Dennard scaling & rise of demanding apps ➔

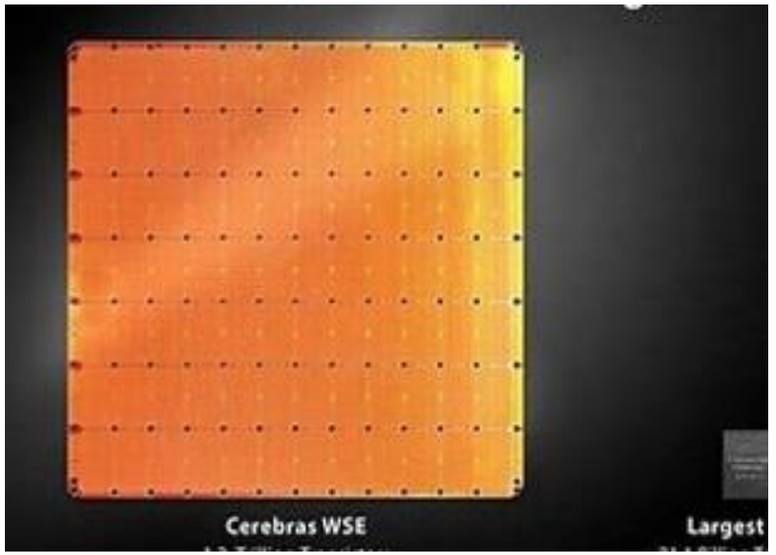
- Accelerator *is a hardware component that executes a targeted computation class faster & usually with (much) less energy.*
- Esp. Deep Neural Network Machine Learning



Nvidia Grace-Hopper



Google Tensor Processing Unit



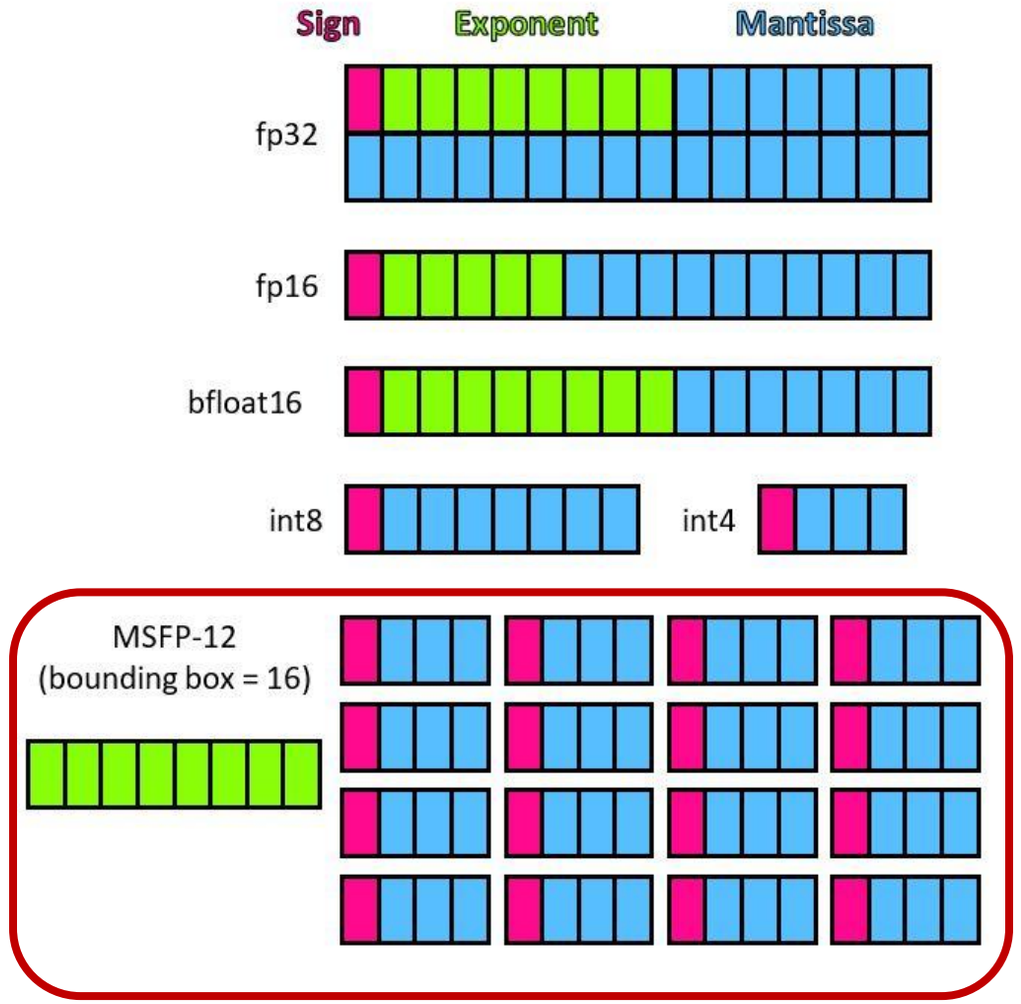
Cerebras Wafer Scale Engine

Compute: Accelerators, Deep Learning Co-design

E.g. Co-Design for Deep Learning via Number Representation

Microsoft FP → Microscaling Formats (MX)

- Mantissa really small
- Multiple values share exponent
- MSFP-12: (8 + 16*4)/16
  = 4.5 bits/value
- Requires co-design



2020: https://www.microsoft.com/en-us/research/blog/a-microsoft-custom-data-type-for-efficient-inference/
2023: https://www.opencompute.org/blog/amd-arm-intel-meta-microsoft-nvidia-and-qualcomm-standardize-next-generation-narrow-precision-data-formats-for-ai

Generative AI

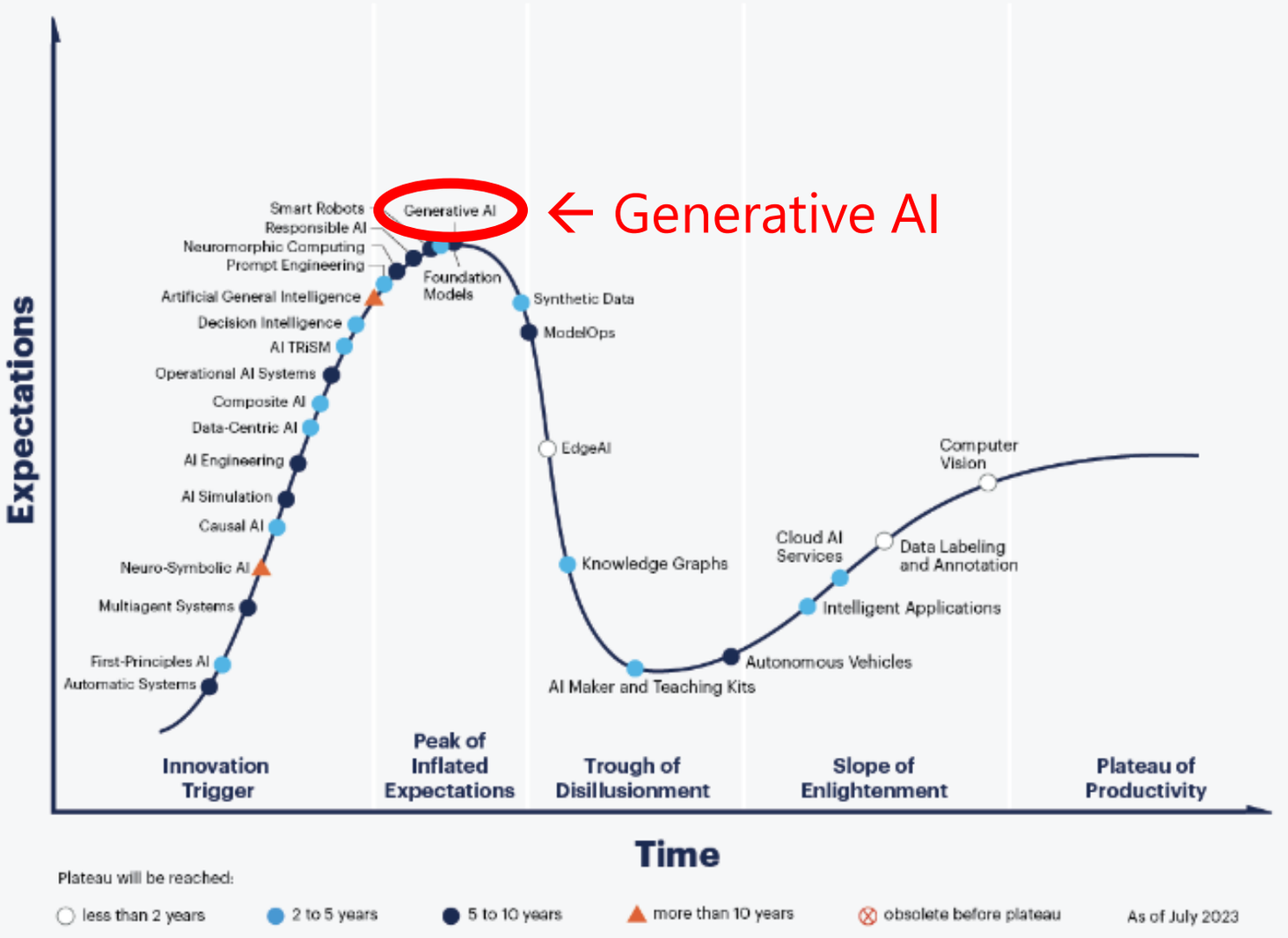
Amazing opportunity: sum >> parts

- Foundation models == means
- Customer value provided by other apps
  - Doing now: make tedious work → faster
  - **Pot of gold: near impossible → practical**
  - Expect bumps: Gartner Hype Cycle (next slide)

- Massive special clusters for foundational AI training: GPUs, TPUs, …
- Growing incremental training. How & where?
- **Exploding inference: wearable, phone, laptop, edge, AND Cloud**
- How to structure AI & GP software and hardware?
- In Cloud, AI clusters will consume massive power → less for GP

New use cases are paramount, & More efficiency → Enables providing value to more people in more ways (see Jevon's Paradox)

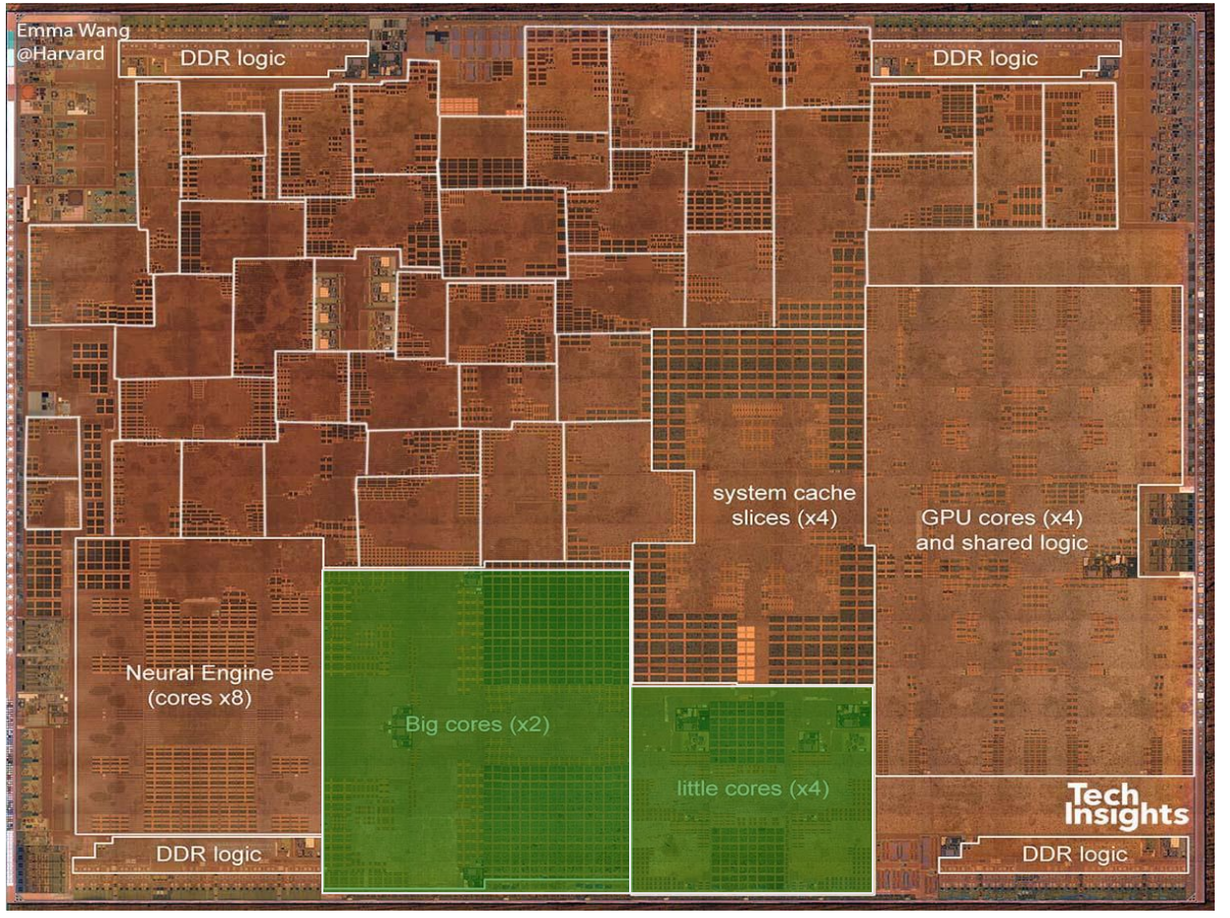
Hype Cycle for Artificial Intelligence, 2023



gartner.com

Gartner

Compute: Accelerator-Level Parallelism



2019 Apple A12 w/ **42 Accelerators**

Deploy Many Accelerators

Use several concurrently

- CPUs: control plane
- Accelerator: data plane

How program, schedule, communicate, co-design?

https://cacm.acm.org/magazines/2021/12/256949-accelerator-level-parallelism

Where to Accelerate?

1st target existing accelerators

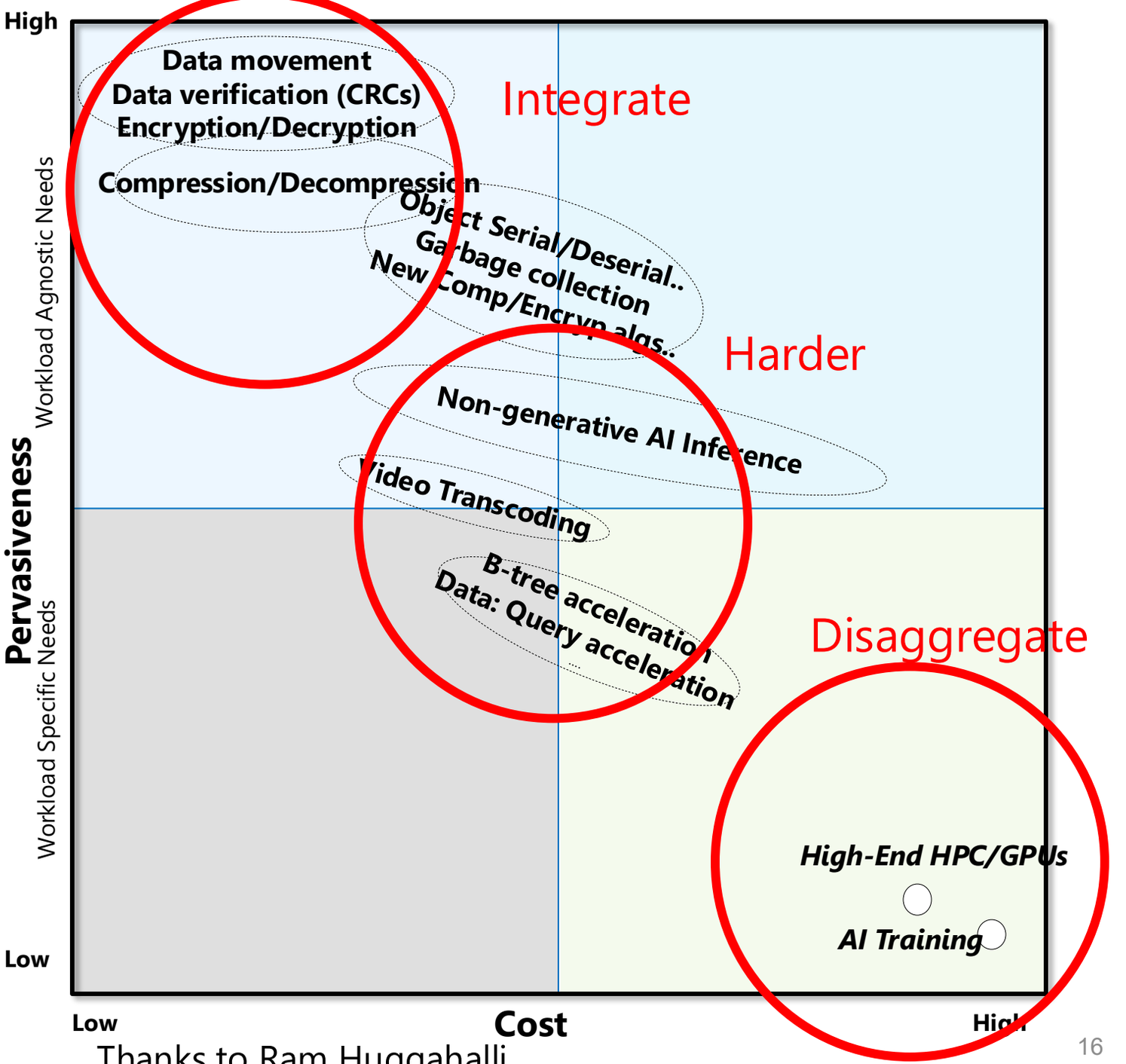
- Data movers & encryption
- Ubiquitous SmartNICs
- Disaggregated GPU/TPUs

Remember

- Amdahl's Law
- Data granularity

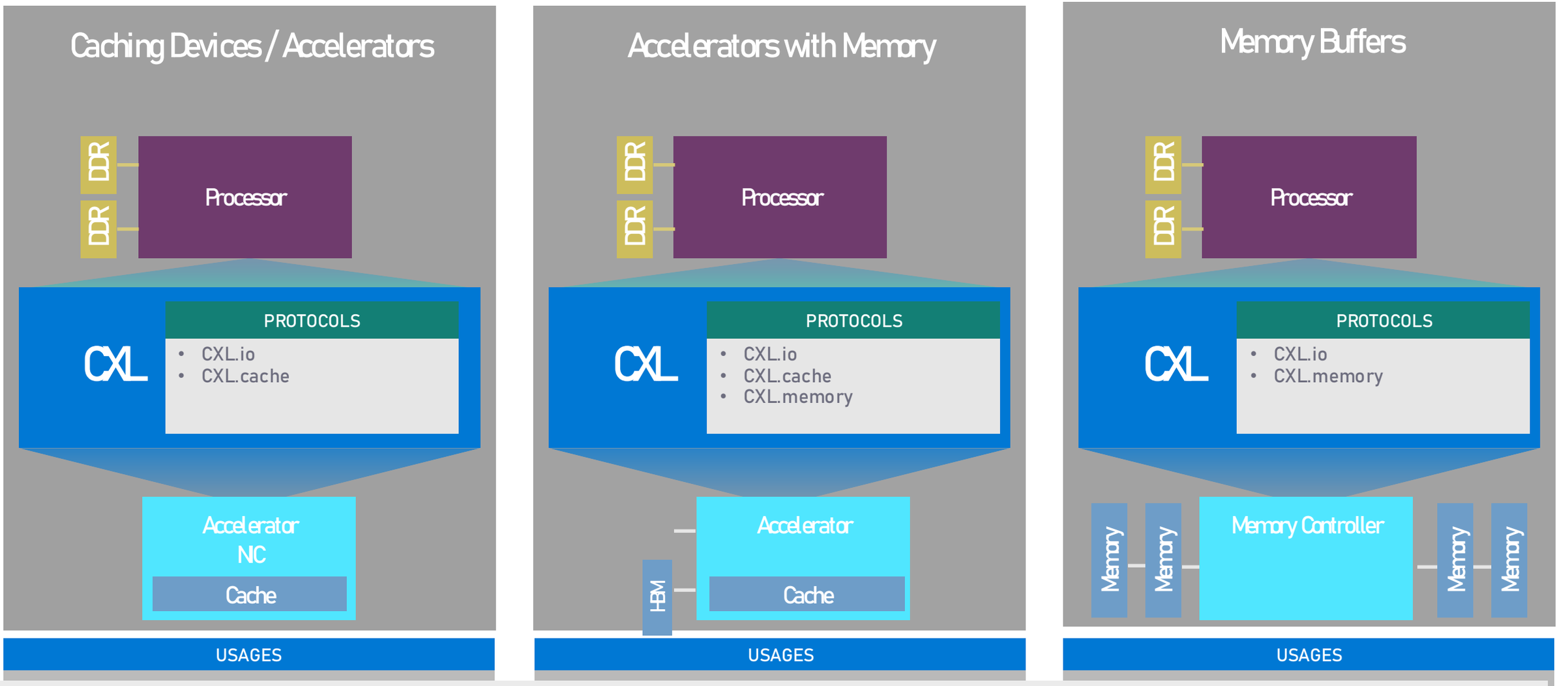
Dedicated accelerators need:

- NRE & design time
- Must provision "right"



Thanks to Ram Huggahalli

New Opportunity: Compute eXpress Link (CXL)



Enables accelerators “closer” than PCIe (coherent) & two-level memory

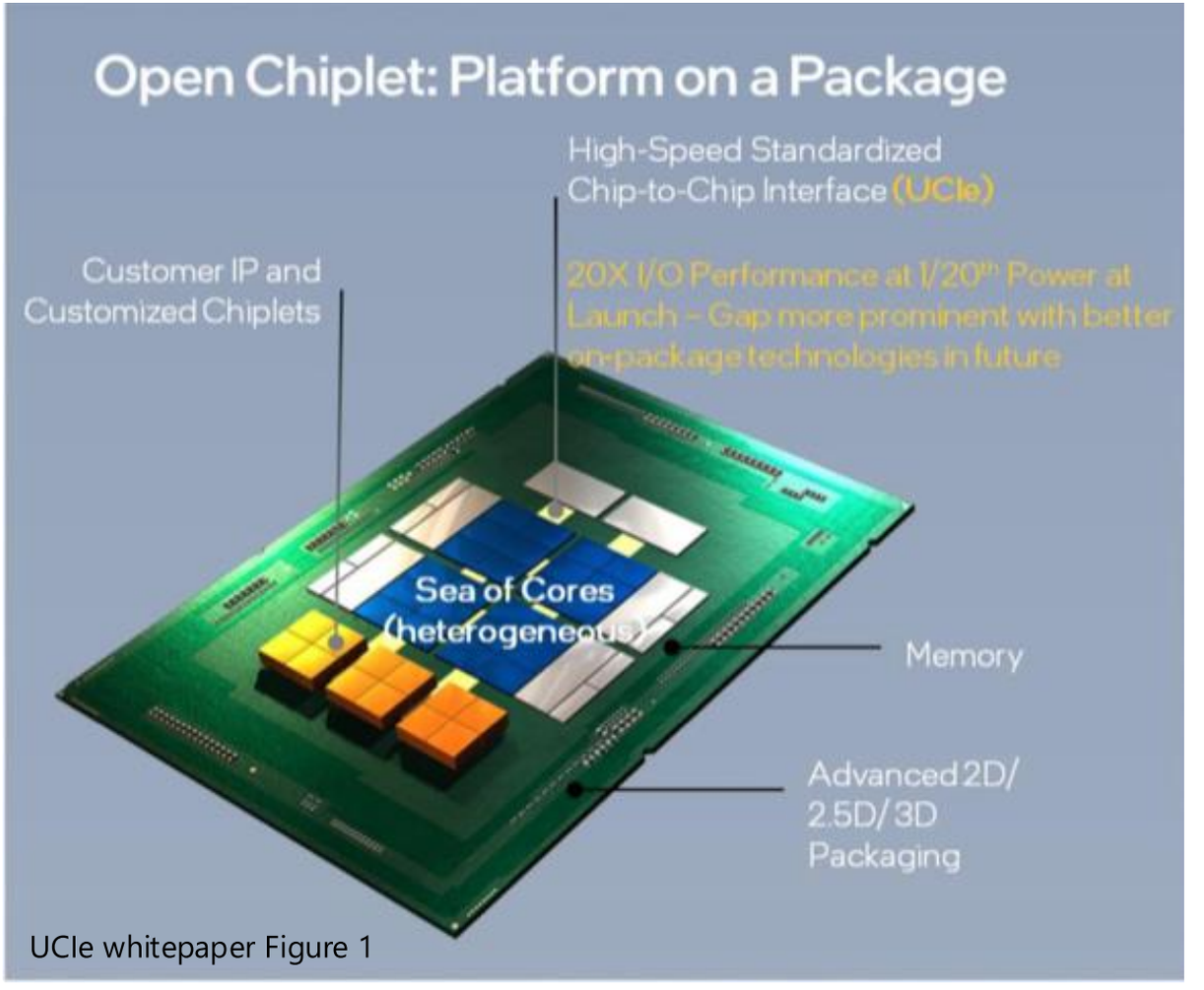
Emerging Opportunity: Universal Chiplet Interconnect Express (UCIe)

Due to Moore's Law Challenges

- Monolithic chip → several "chiplets"
- Fast Silicon interconnect
- Currently company proprietary

Emerging UCIe Standard

- Make package like a "board"
- Standardized protocol among chiplets (physical/electrical/link/transport)
- Get closer: PCIe > CXL > UCIe
- Mix/match chiplets from different technologies/companies
- https://doi.org/10.1038/s41928-024-01126-y



UCIe whitepaper Figure 1

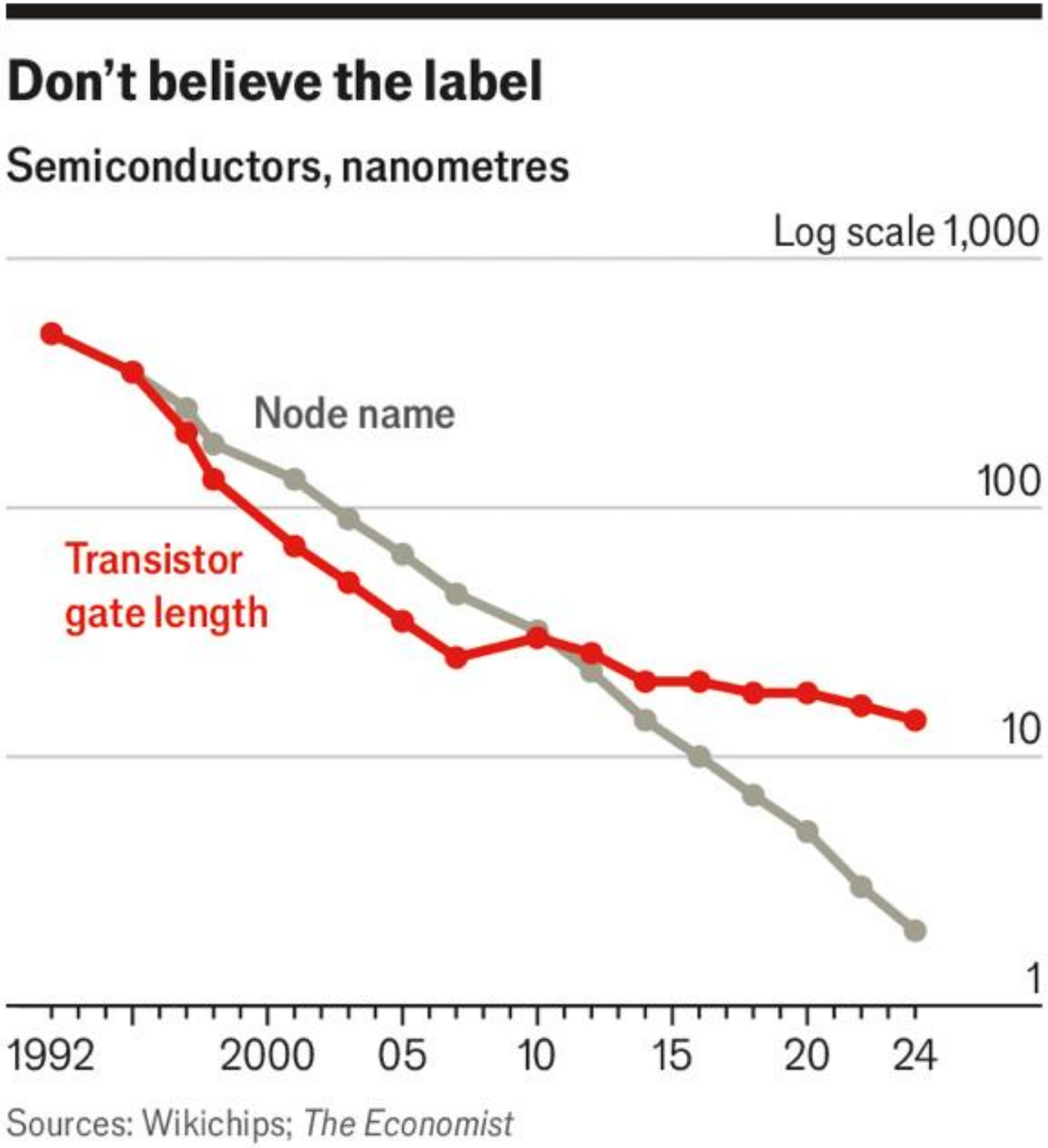
2D then 2.5D then 3D. Why 3D?

Tech Scaling Frontier from 2D To 3D & Advanced packaging

Why? 2D scaling slowing

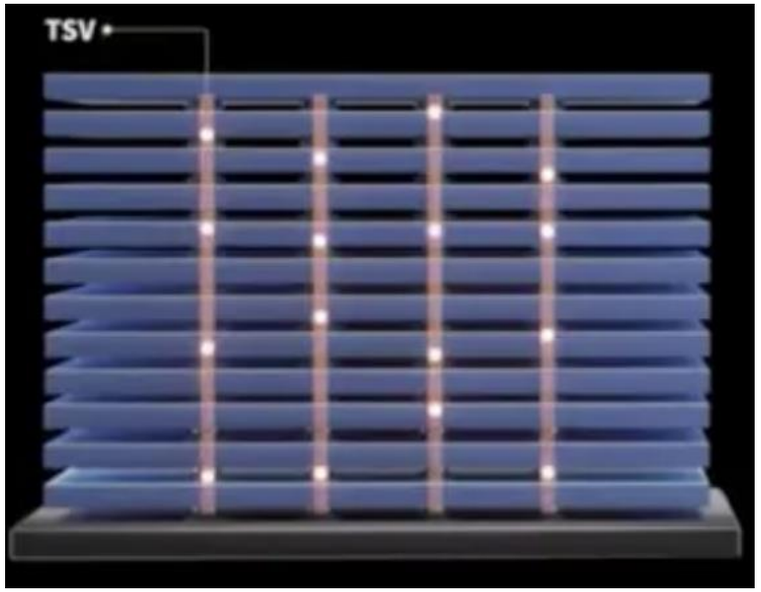
2D then 2.5D then 3D

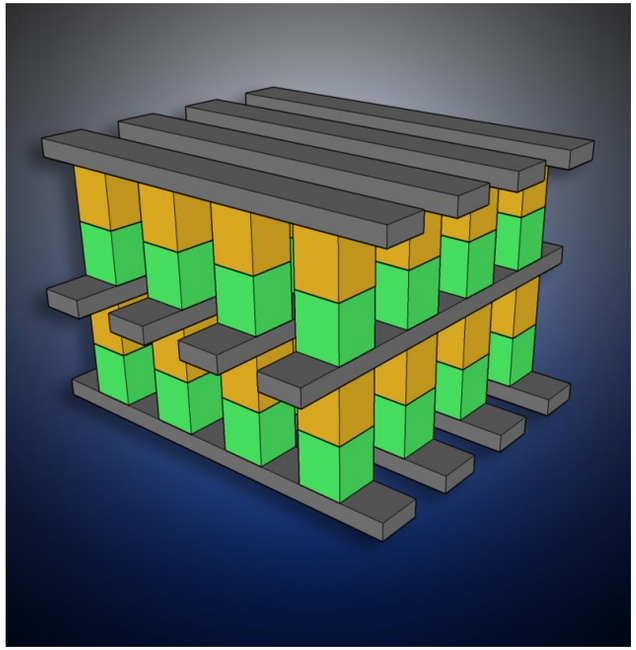
What does 3D look like?

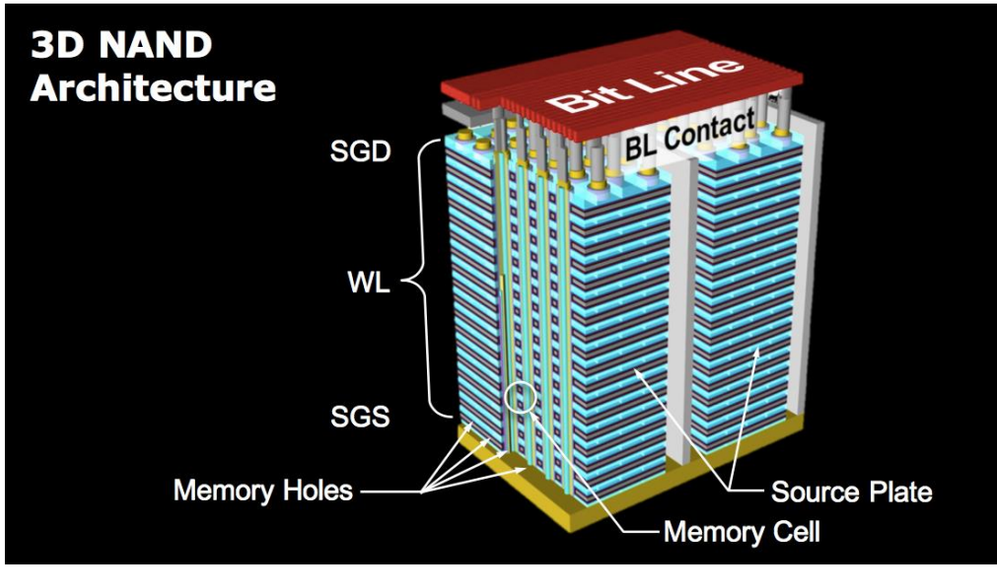


The Economist, Technology Quarterly: Chipmaking, 21 September 2024

Three Ways for 3D Scaling (to continue “perf” Moore’s Law)







Fab 2D chips; stack with TSVs

(high BW memory)

Works; expensive

Fab “Decks” that stack 3D

(Intel Optane)

Tricky; medium cost

Fab real 3D circuits

(NAND FLASH)

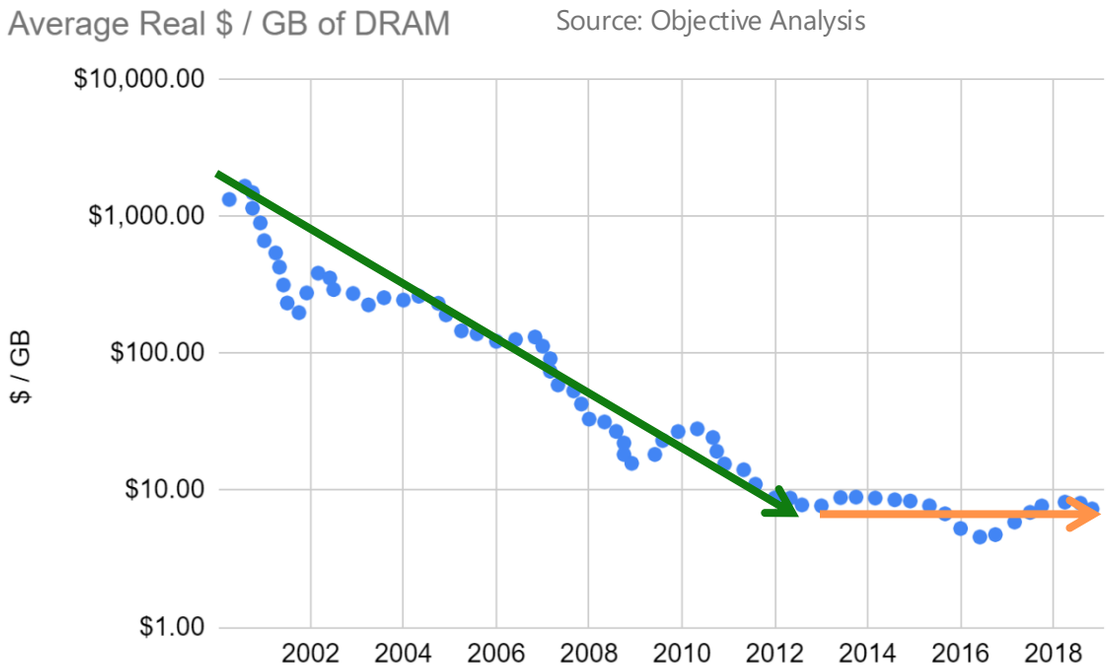
Holy Grail but difficult

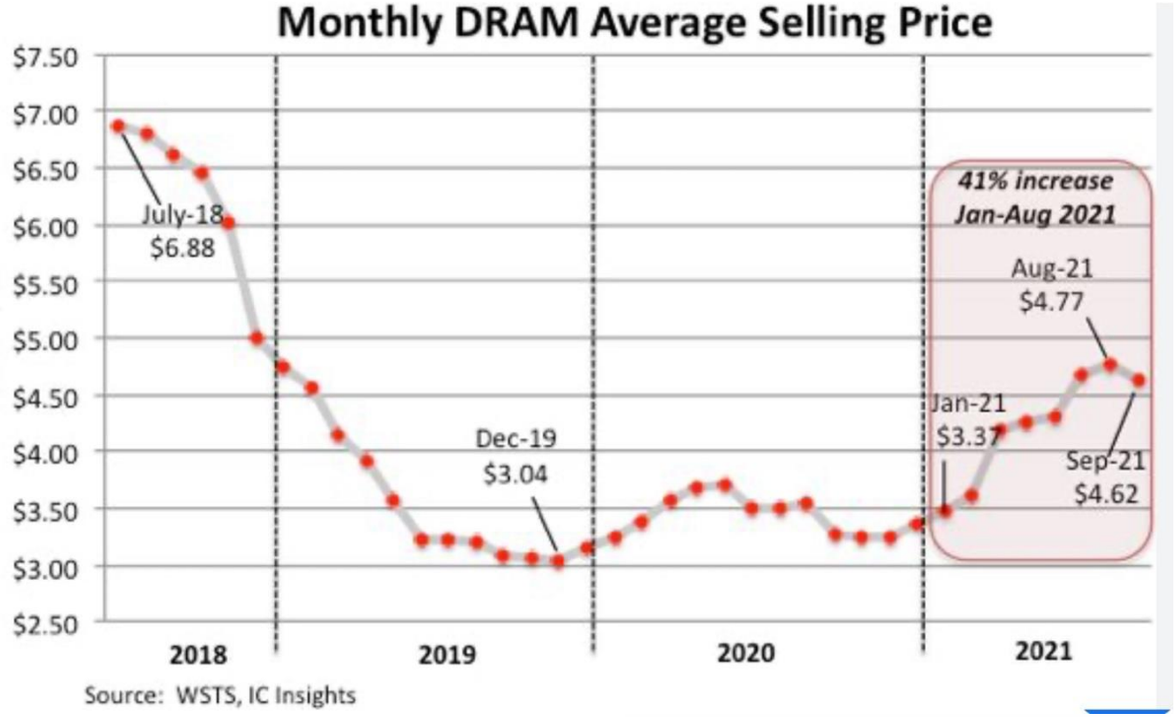
Computer Architecture's Eternal Questions & Outline

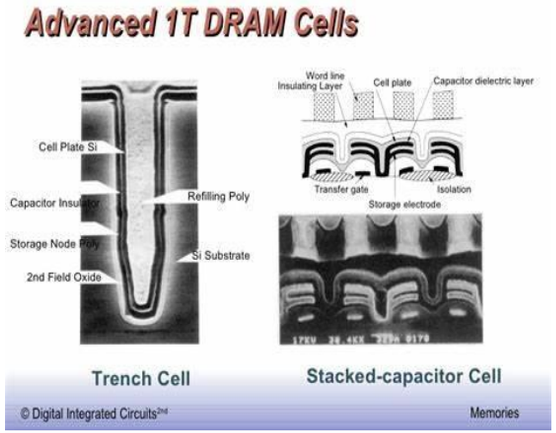
How best to do these interacting factors:

1. ~~**Compute (longest)**~~
2. **Memory (longer)**
3. Interconnect/networking
4. Storage
5. Security
6. Power
7. Cooling
8. *Bonus new question*

Memory: Vast, Fast, Synchronous DDR ➔ Untenable







DDR DRAM price not scaling ➔ poor 2D scaling
➔ With DDR only, future cores/socket growth slows

Force Response: Two-Tier Memory (c.f., Multicore)

CXL Type 3 enables two-level memory

Extended Memory w/ What Tier 2 tech?

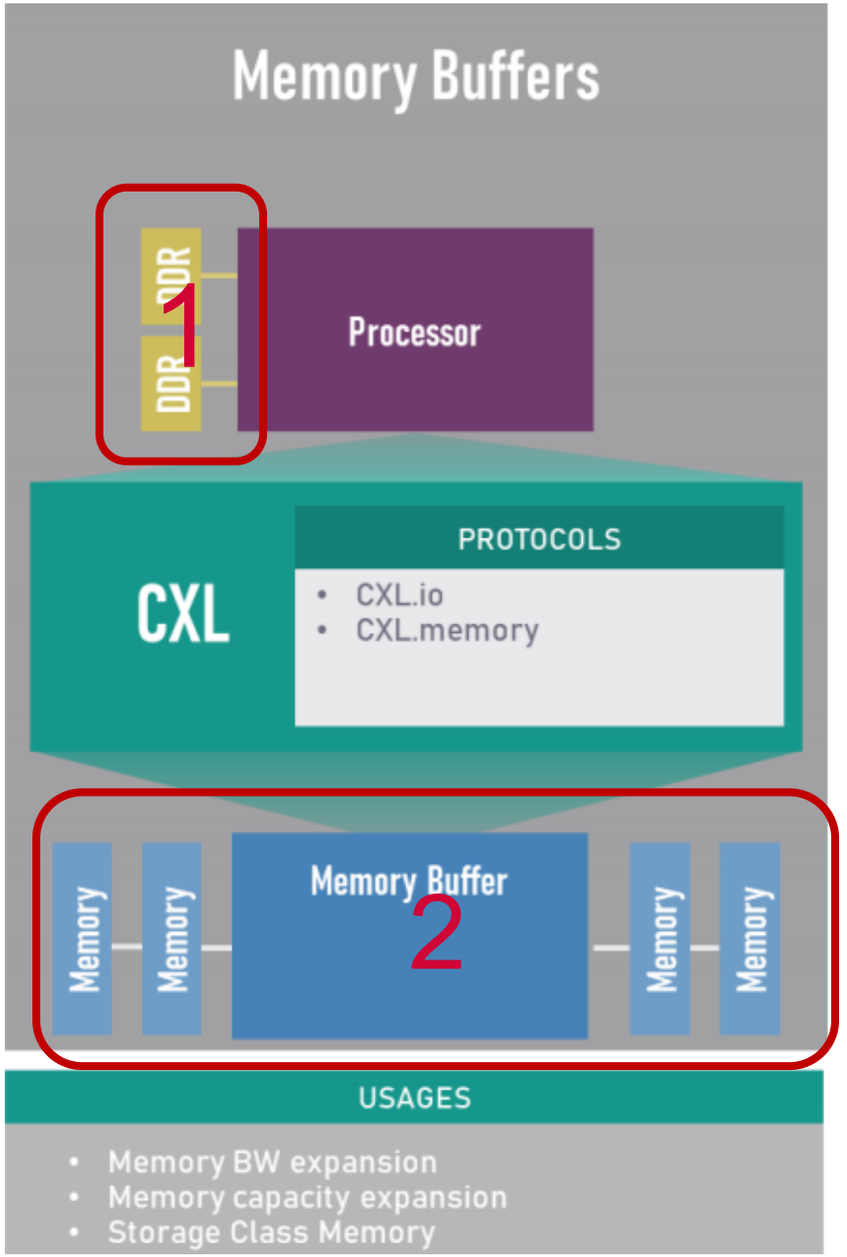
- More DDR5
- Emerging Memory Technologies

How manage?

1. Auto-HW, e.g., Intel Flat Memory Mode

   E.g., Managing Memory Tiers with CXL … [OSDI 2024]
2. Hypervisor Managed with latency-sensitive pages in DDR5
3. Application Aware (Explicit)

   E.g. Two-level database buffer pool



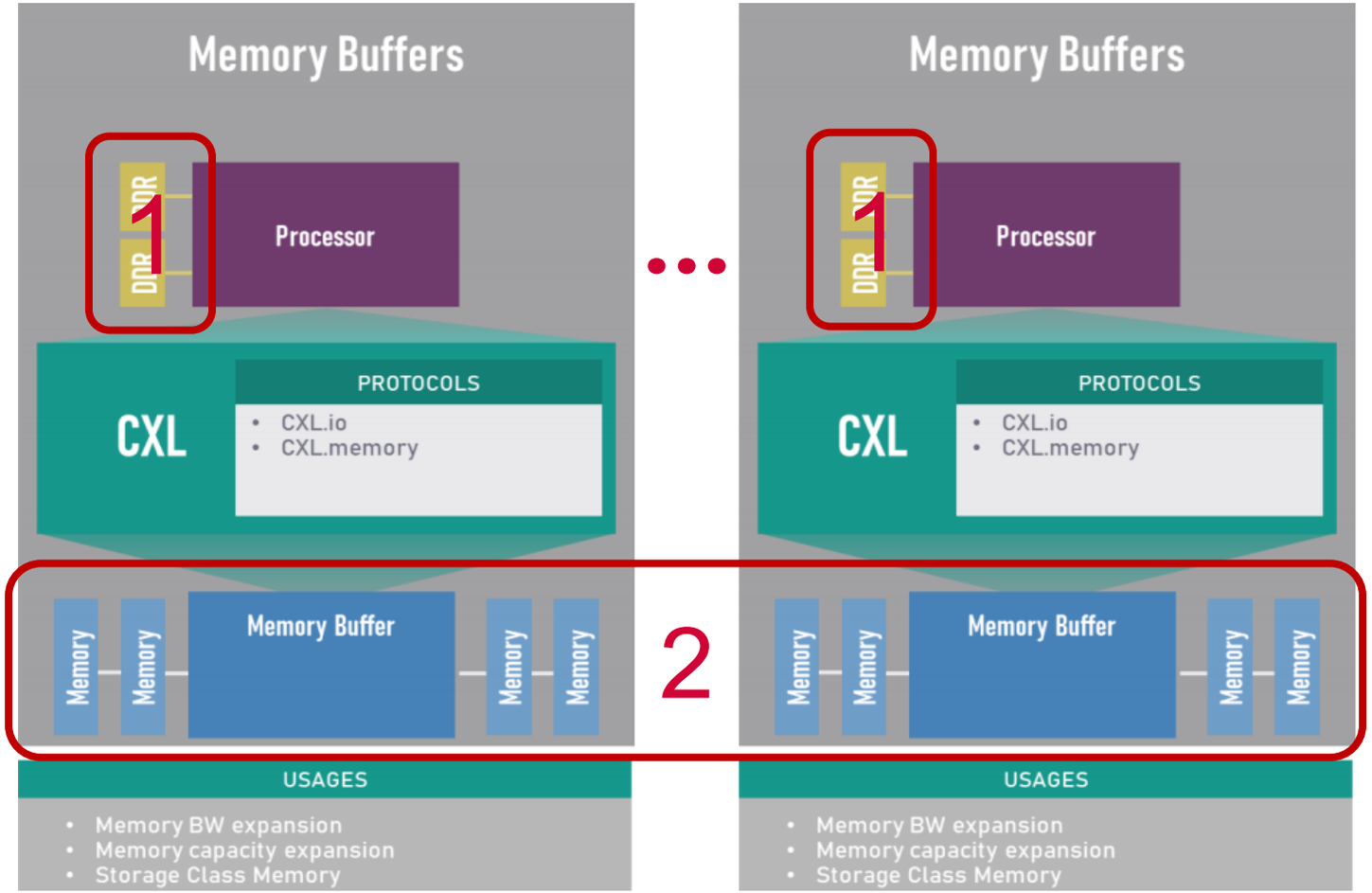
After CXL extended memory: Pooling & Sharing

Many-socket HW coherence support withering. What about analytic databases?

CXL Opportunity

- Connect several sockets to same CXL memory

1. **Pooling**: each region attached by one socket at a time
   E.g., Pond pooling [ASPLOS'23]
2. **Structured Sharing** with limited HW coherence
   E.g., Tigon: Distributed DB [OSDI'25]



Memory: Processing In Memory (PIM)

Usually, move all data to CPU(s)

PIM: Move compute to vast data in memory

- A high pain, high grain opportunity

Old idea revived by

1. Conventional compute's energy problems
2. Important apps: Deep Learning & Recommendation
3. Attention from serious memory vendors

Alternatives: Processing {In, Near} Memory

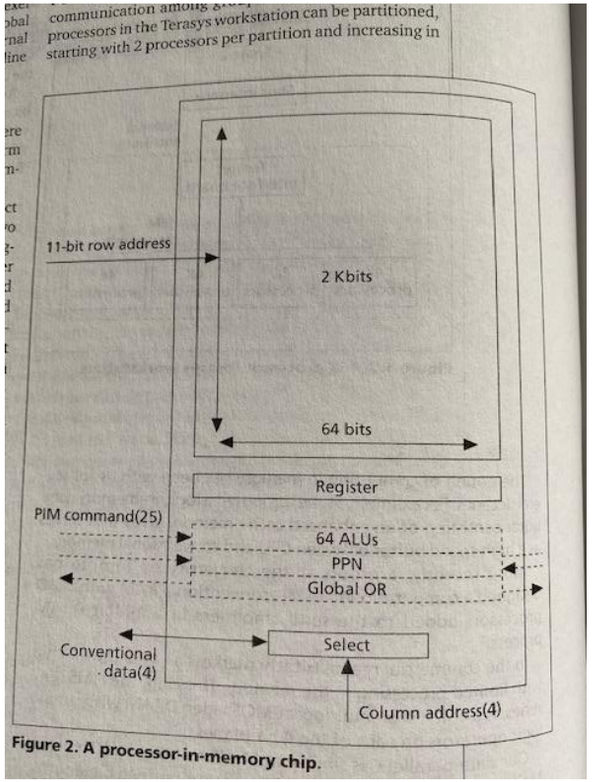


Figure 2. A processor-in-memory chip.

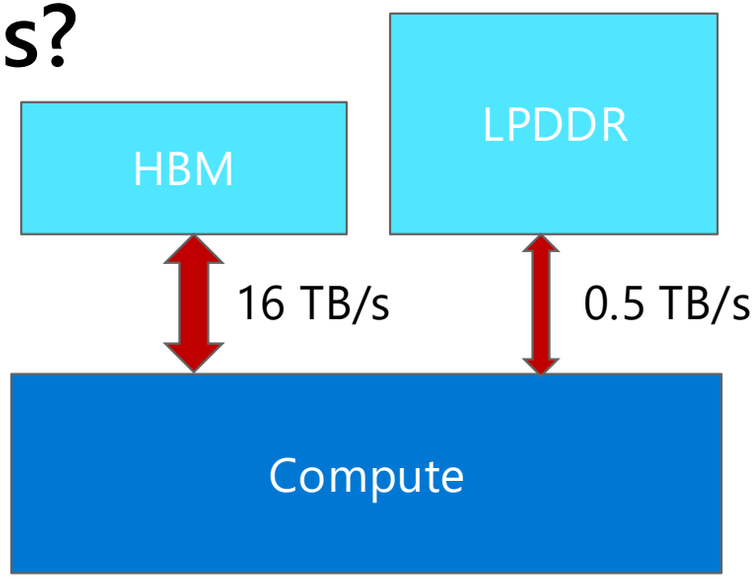
Gokhale,Holmes,Iobst [1995]

Hardware Architecture and Software Stack for PIM
Based on Commercial DRAM Technology
Sukhan Lee, et al., **Samsung**, ISCA Industrial Track, June 2021

Processing NEAR memory
as more likely IMHO, e.g.,
logic die under HBM stack

Whither AI (Inference) Memory Hierarchies?

Model fits in HBM; KV Cache does NOT;
With PagedAttention, fetch multi-MB blocks



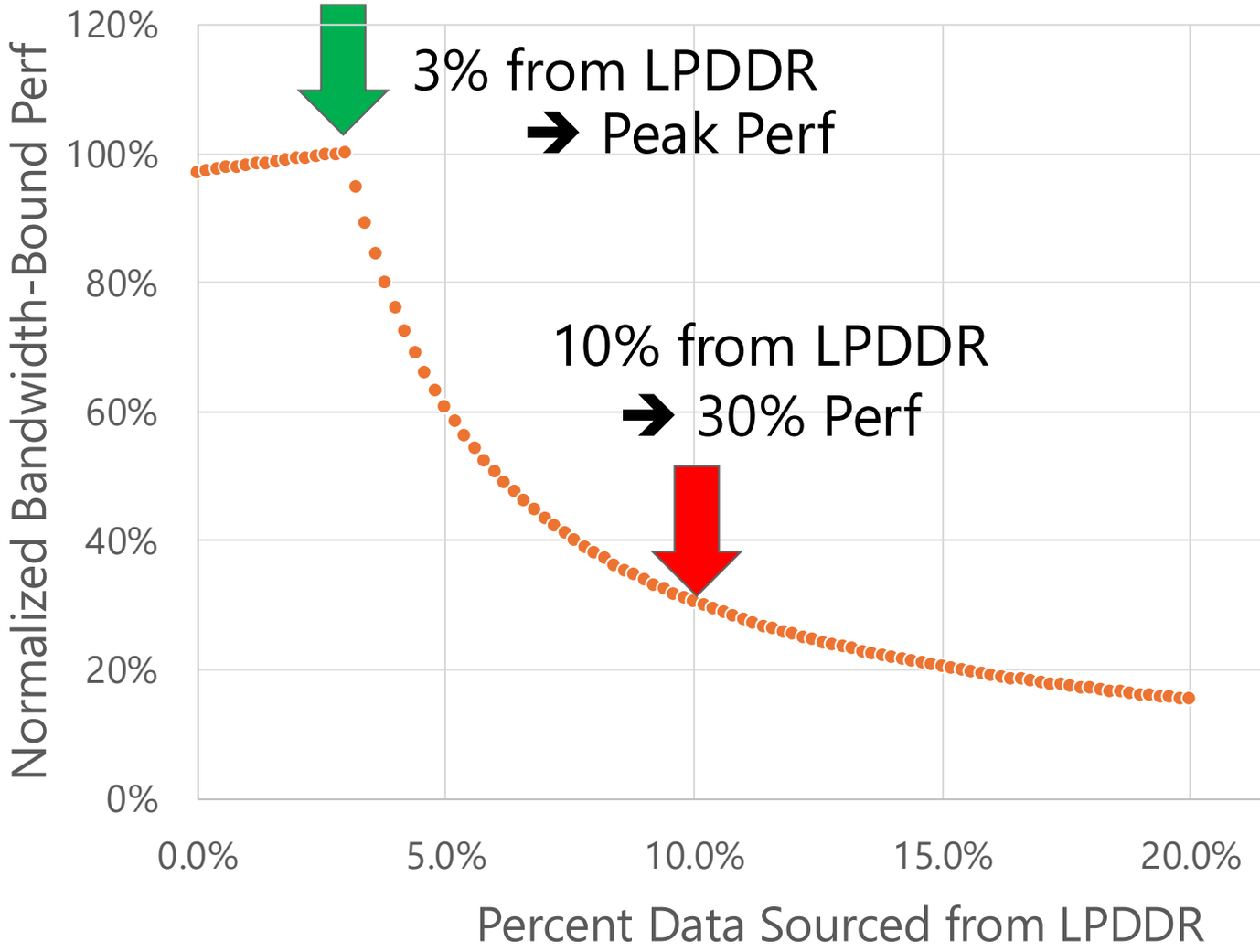
Performance determined by Bandwidth!

Consider 100% bandwidth-bound extreme – computation hidden

- 16 TB/s HBM bandwidth
- 0.5 TB/s LPDDR bandwidth

- What fraction of data from LPDDR yields best performance?

Answer: Have LPDDR source 3% Data



Why?

Want LPDDR & HBM data transfers to end together

LPDDR provides 3% BW = 0.5/(16.0 + 0.5)

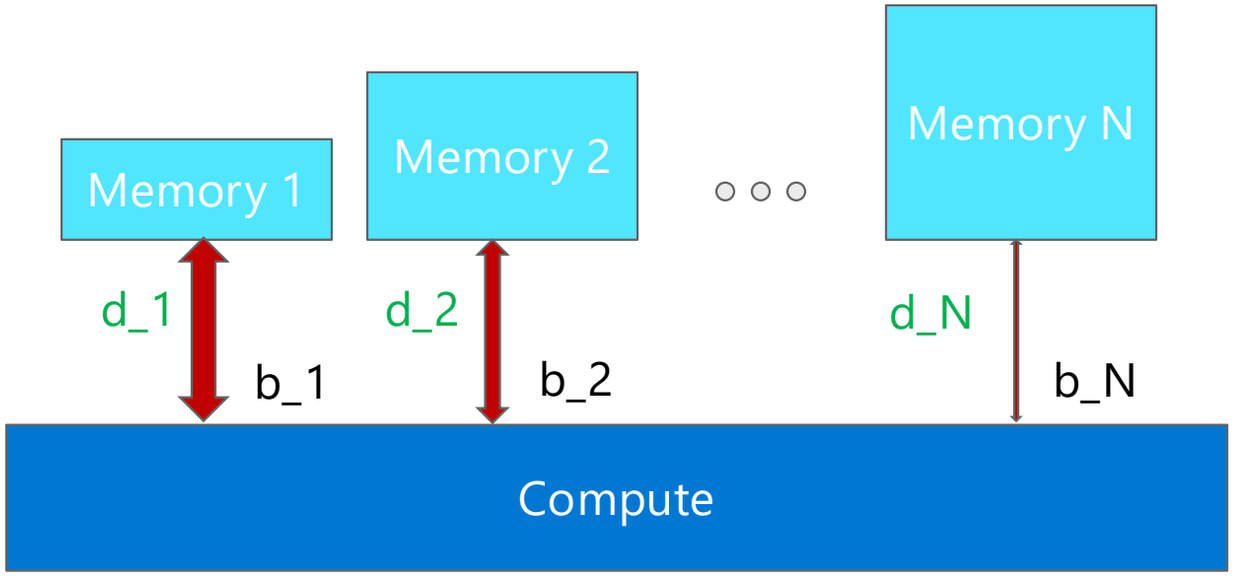
Have LPDDR transfer 3% of needed data

Is HBM size sufficient to hold 97% of needed data?

What if N memory levels?

What if caching allowed?

Whither N Memory Levels?



Principle (without caching):

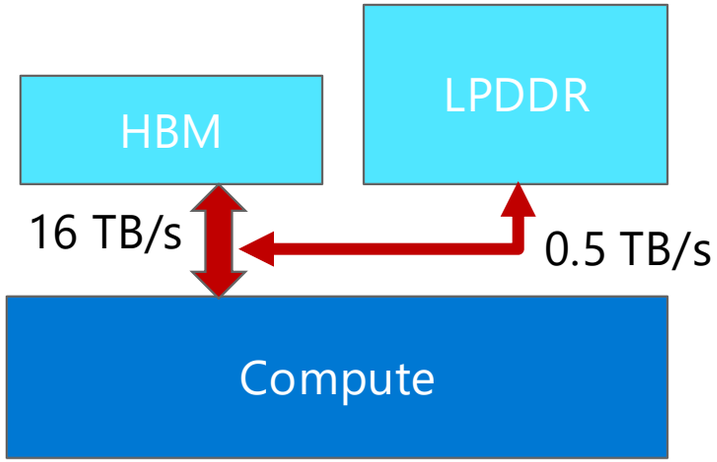
Memory-BW-bound workloads maximize performance when they source data from each memory level proportional to its bandwidth

Perhaps known to HW people; good & simple for SW people

Memory hierarchies determined by BW ≠ Classic CPU ones

Work in Progress

Impact of "Caching" LPDDR Data in HBM



Let ALL LPDDR data transfer to HBM first
HBM provides all data to compute

Both at 97% using HBM BW only

Direct transfer peaks at 3% using HBM & LPDDR BW perfectly in parallel

Both decline rapidly when moving too much data with limited LPDDR BW, and it doesn't matter if LPDDR data goes to HBM or not

Caching declines slowing moving as LPDDR data to HBM uses some HBM BW

KEY: Caching moves location left on x-axis whenever one LPDDR transfer enables many HBM transfers

4% = 5x Caching Reuse

20% Direct

Normalized Performance

120.0%
100.0%
80.0%
60.0%
40.0%
20.0%
0.0%

0.0% 0.6% 1.2% 1.8% 2.4% 3.0% 3.6% 4.2% 4.8% 5.4% 6.0% 6.6% 7.2% 7.8% 8.4% 9.0% 9.6% 10.2% 10.8% 11.4% 12.0% 12.6% 13.2% 13.8% 14.4% 15.0% 15.6% 16.2% 16.8% 17.4% 18.0% 18.6% 19.2% 19.8%

Percent Data in LPDDR

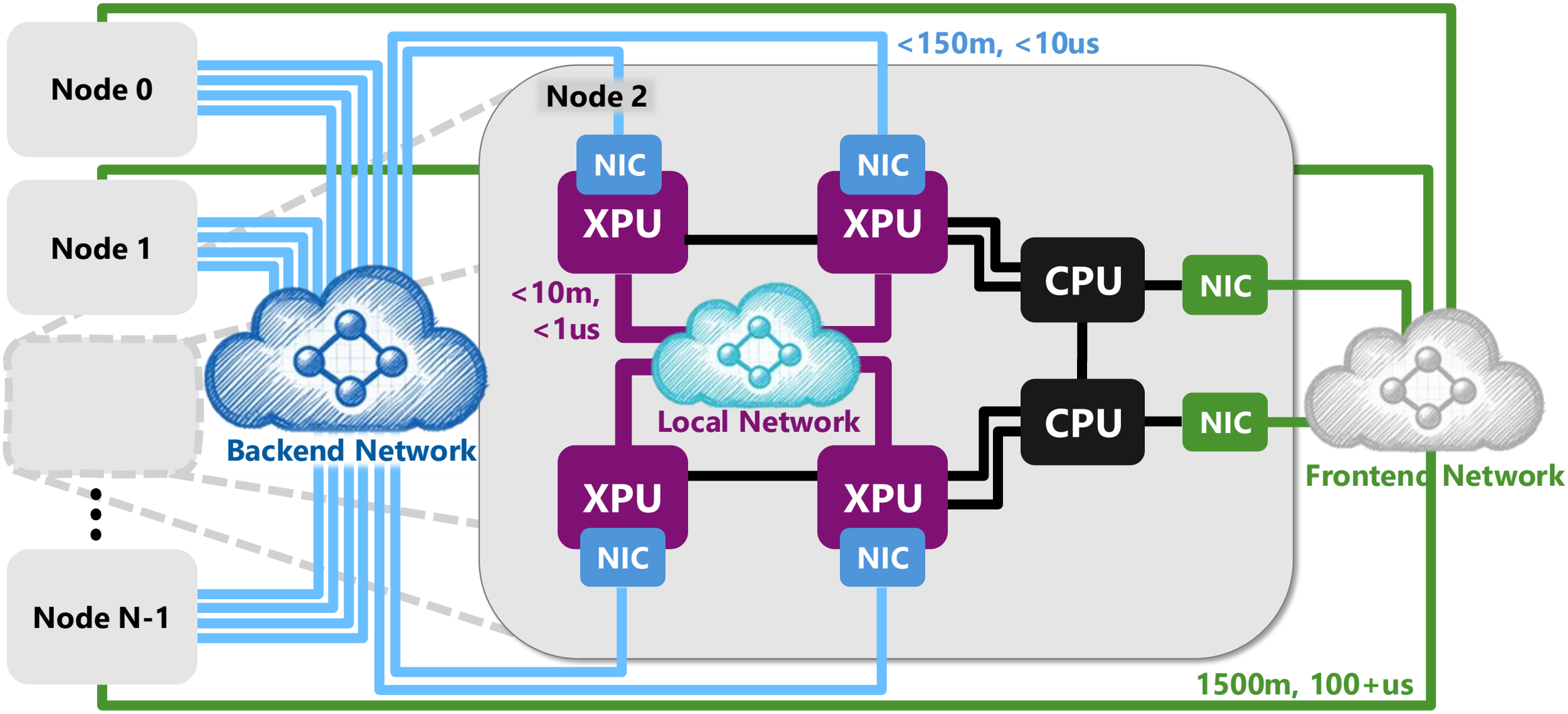
Direct Transfers Cache in HBM

Computer Architecture's Eternal Questions & Outline

How best to do these interacting factors:

1. ~~**Compute (longest)**~~
2. ~~**Memory (longer)**~~
3. Interconnect/networking ← CXL & UCIe already covered
4. ~~Storage~~
5. Security
6. Power
7. Cooling
8. *Bonus new question*

Three Data (AI) Center Networks (https://arxiv.org/abs/2508.08906)



Ultra Ethernet (https://ultraethernet.org/)

Started by Alphabet, AMD, Arista, Atos, Broadcom, Cisco, HPE, Intel, Meta, Microsoft, Oracle, but many more now

Goal: Provide a high-performance low-cost Ethernet-based solution for emerging AI and other high-bandwidth low-latency workloads

Insight: Improve Ethernet by focusing

- **Workloads in a data center**
- **Rather than arbitrary Internet**
- **Most changes to transport layer (beyond TCP)**

Insight: Use smarter NICs as compute improving faster than BW

Targeted solutions: connectionless packet delivery, packet spraying, optional relaxed ordering, phase-aware congestion control, & more

Multiple switches & network interface cards (NICs) under development

Security: Confidential Compute (CC)

Cloud Providers Now:

- Promise to protect your data/code from outsider/insider threats

With Confidential Compute

- **Your data/code is cryptographically protected from both threats**
- Hard: Root of trust, attestation, inter-package comm encrypted, memory/storage w/ data/address/replay protected, …
- Can expand markets, but correctness/efficiency challenges

CC: https://queue.acm.org/detail.cfm?id=3456125 [ACM Queue'21] & ACM Queue Jul/Aug'23 issue
OpenSource Root-of-Trust: https://petri.com/microsoft-caliptra-open-source-root-of-trust/
Azure Sphere (IoT): https://aka.ms/7properties

New ideas & accelerators must be compatible with CC.
E.g., accelerator or switch trusted to manage tenant crypto keys?

Power: IoT to Cloud Varies

Wearables/IoT/Mobile: Energy (battery life)

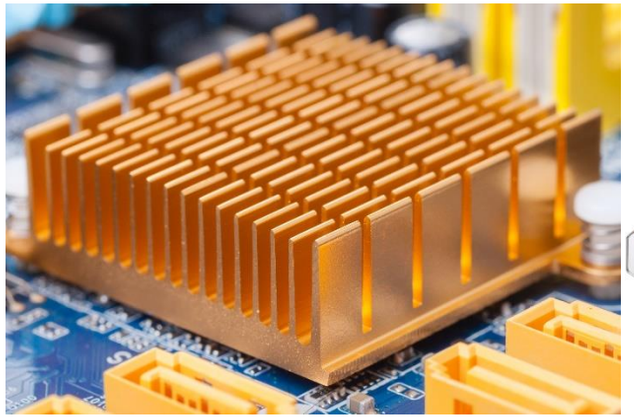
- Save energy: Use little energy ~idle
- Add energy: E.g., harvesting

Cloud: Constant Power

- Mega-datacenters pay for fixed power
- **Using less power doesn't save money**
- **How to use constant power well?**
- Intermittent, renewable power expanding
- MSFT contracts w/ Helion Fusion [5/2023] & Three-Mile Island Fission Plant [9/2024]

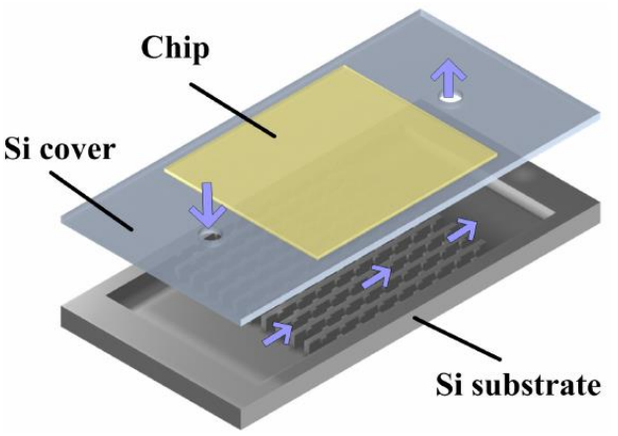
https://www.microsoft.com/en-us/research/uploads/prod/2020/10/Per-VM-Capping-ATC21.pdf

Cooling

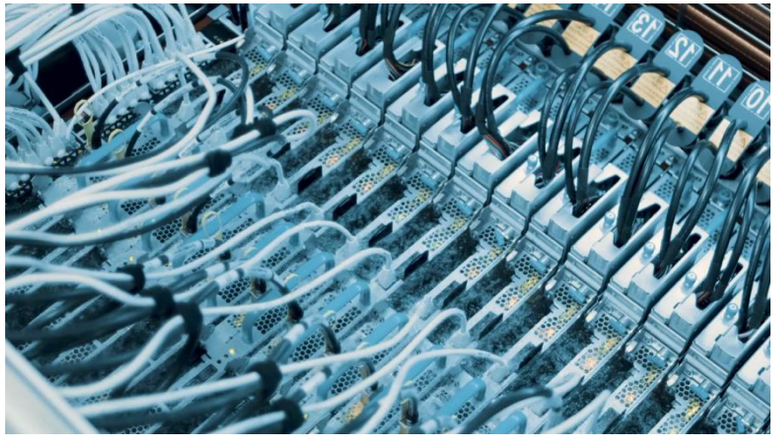


Air w/ heat sink

Cold plate



Microfluidics



Immersion (1 or 2 phase)???

Data Centers are becoming gigantic supercomputers!

How might these interact with computer architecture's other eternal questions?

https://news.microsoft.com/innovation-stories/datacenter-liquid-cooling/

Exploit compact access to vast compute, memory, & storage?

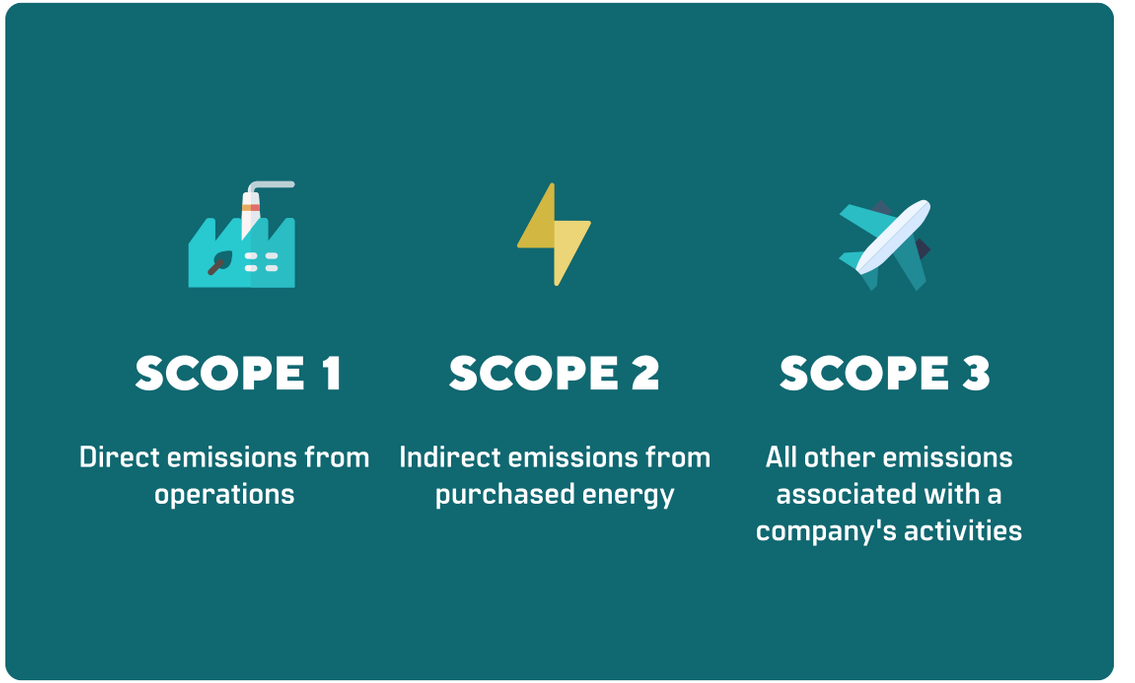
(Bonus) Sustainability!

How to reduce provisioned power (scope 2) & Si area (scope)? 3)?

I said comp arch's questions don't change but
George Box: *All models are wrong, but some are useful.*

New: Make Computing More Sustainable?

Green House Gas Emission Scopes



US EPA: https://www.epa.gov/ghgemissions

Microsoft seeks carbon negative by 2030, https://www.microsoft.com/en-us/corporate-responsibility/sustainability

See also Harvard & Facebook/Meta HPCA 2021 (https://ieeexplore.ieee.org/document/9407142/)
& ISCA 2022 (https://dl.acm.org/doi/epdf/10.1145/3470496.3527408)

Computer Architecture's Eternal Questions & Outline

How best to do these interacting factors:

1. ***Compute:*** accelerators, deep learning, & many
2. ***Memory:*** 2D scaling dead & processing near memory
3. ***Interconnect/network:*** protocols/optics
4. ***Storage:*** mind the gaps
5. ***Security:*** confidential compute
6. ***Power:*** IoT to cloud varies
7. ***Cooling:*** consider cold plate & its impact
8. New: ***Sustainability:*** whither emission scopes 1, 2, & 3?