Taming the Beast

Efficiency in an AI/Crypto World

Bill Gervasi, Principal Memory Solutions Architect

Monolithic Power Systems

bill.gervasi@monolithicpower.com

the Future of Memory and Storage

Reuters

America's largest power grid is struggling to meet demand from AI

Electricity bills are projected to surge by more than 20% this summer in some parts of PJM Interconnection's territory, which covers 13...

Financial Times

Hitachi Energy says AI power spikes threaten to destabilise global supply

Big Tech's spiking electricity use as it trains artificial intelligence must be reined in by governments in order to maintain stable...

Tom's Hardware

AI is eating up Pennsylvania's power, governor threatens to pull state from the grid — new plants aren't being built fast enough to keep up with demand

Spiking demand is sending energy bills skyrocketing, while the governor threatens to pull the state from the grid.

Yahoo Finance

AI power demand poses global supply risks, says Hitachi Energy

In an interview with the Financial Times, Hitachi Energy CEO Schierenbeck urged government action on AI's unpredictable power demands.

EES2
ENERGY
EFFICIENCY
SCALING

Designing for energy efficiency is a growing concern

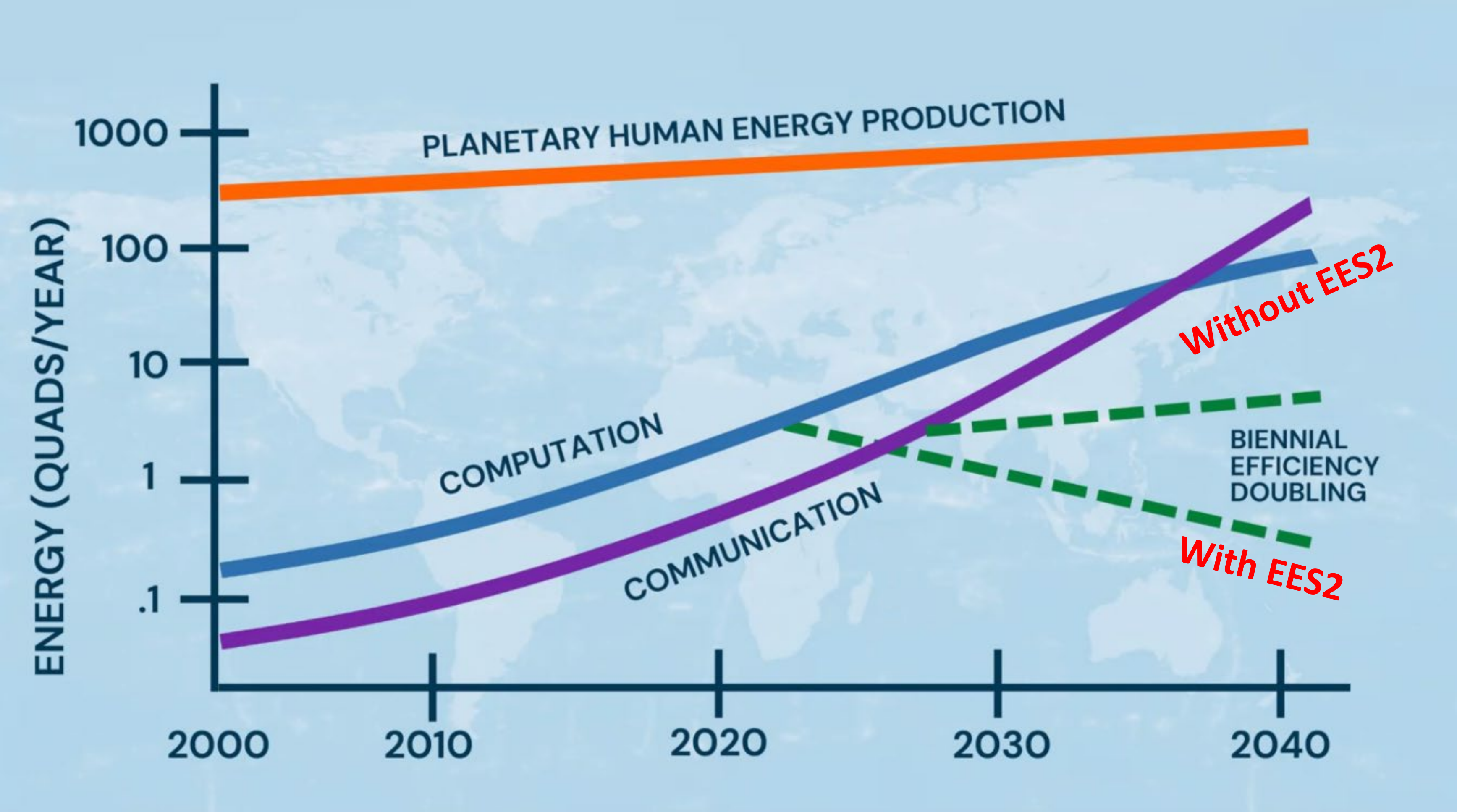
MPS

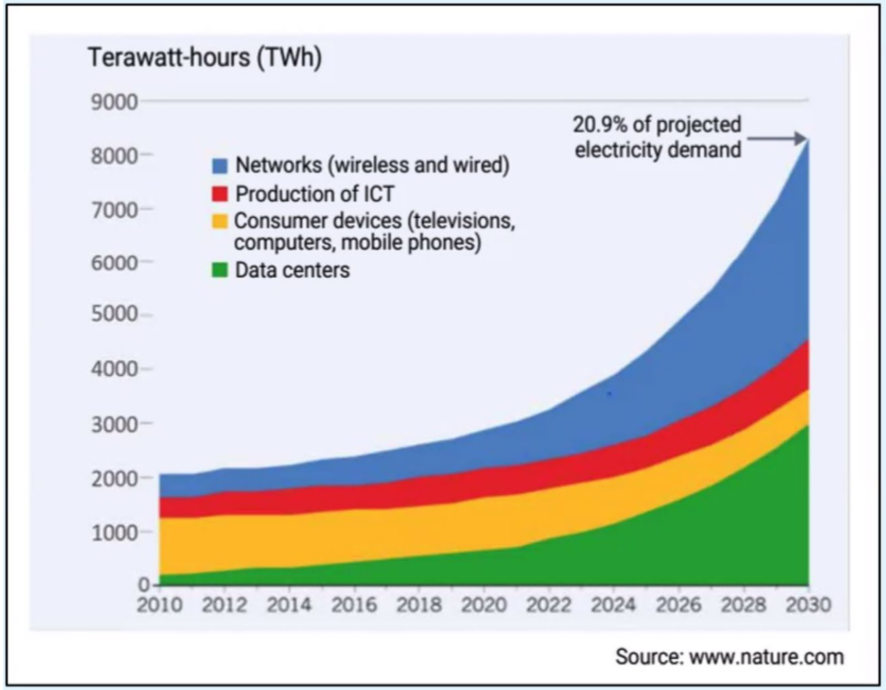
On the current trajectory of energy use versus energy production,

THESE CROSS OVER IN 2055

EES2 program goal is 1000X improvement in energy efficiency over the next 20 years

This program is not US-centric
All countries are invited to participate





Operation	Energy per bit
Wireless data	10 – 30μJ
Internet: access	40 – 80nJ
Internet: routing	20nJ
Internet: optical WDM links	3nJ
Reading DRAM	5pJ
Communicating off chip	1 – 20 pJ
Data link multiplexing and timing circuits	~ 2 pJ
Communicating across chip	600 fJ
Floating point operation	100fJ
Energy in DRAM cell	10fJ
Switching CMOS gate	~50aJ – 3fJ
1 electron at 1V, or 1 photon @1eV	0.16aJ (160zJ)

most energy is used for communications, not logic

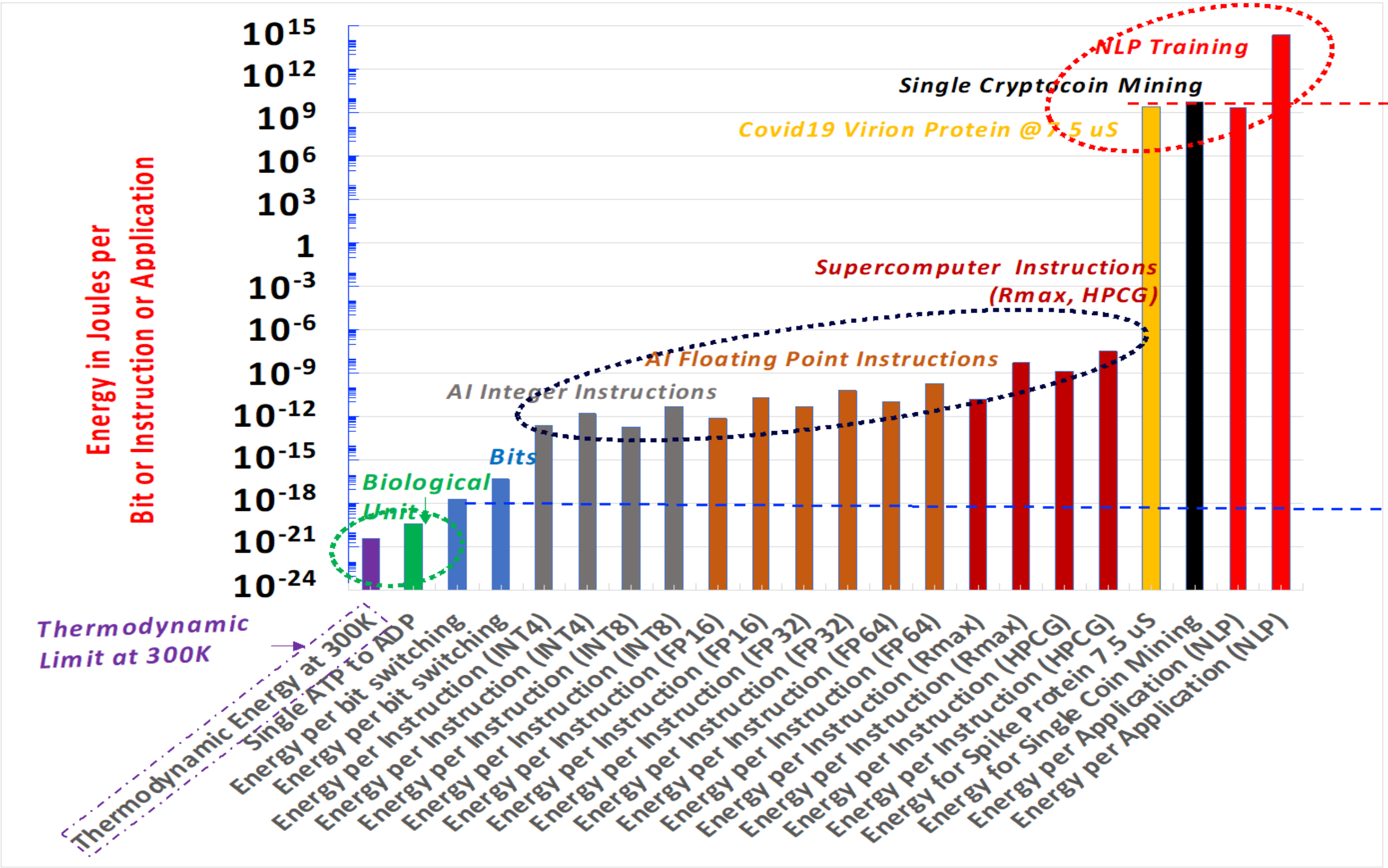
You can't solve a problem if you can't name it

EES2 Phase 1 report identified where we are spending power

Also, what technologies can improve efficiency

Phase 2 began in June 2025 to initiate action

Conclusion: we are better at moving data around than we are at operating on that data



10^{9} to 10^{15}

10^{-18} to 10^{-21}

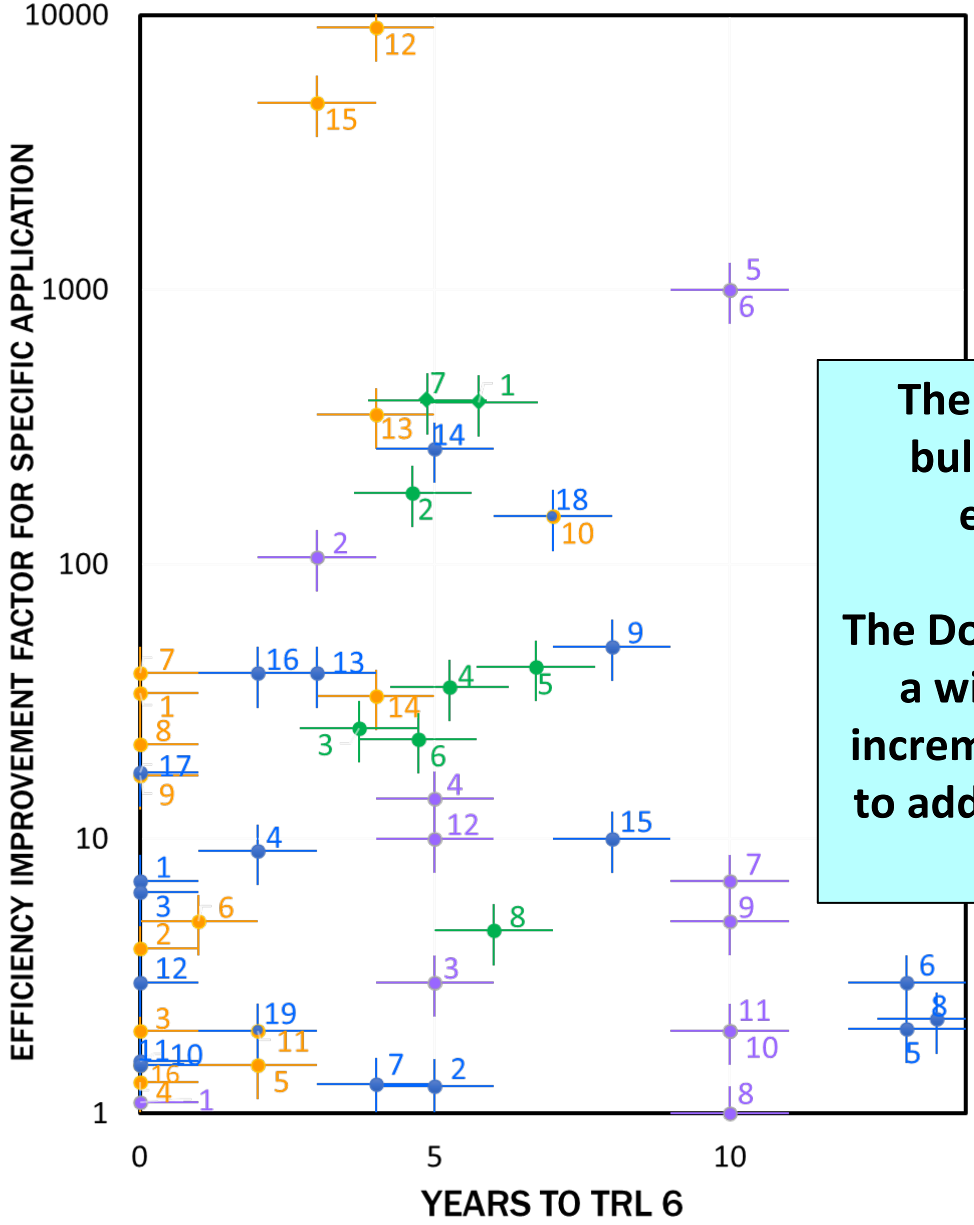
27 to 36 orders of magnitude
(From Algorithms to Atoms)

Part of the looming energy crisis is fundamental inefficiencies of applications and programming languages

Python programming is orders of magnitude less energy efficient than C programming (ChatGPT is Python-based)

Cryptocurrency in particular consumes ≥0.8% of world energy resources already

the Future of Memory and Storage



There is no silver bullet that fixes everything

The DoE has identified a wide variety of incremental solutions to address the power crisis

TRL: Technology Readiness Level

Circuits and Architectures

1 ReRAM vs NAND
2 STTRAM vs NAND
3 NRAM vs DRAM
4 ReRAM vs DRAM
5 CNT NVM
6 Metis SRAM
7 Molecular dynamics ASIC
8 FPGAs for machine vision
9 SRAM stacked 3D DNN accelerator
10 MIV stacked ReRAM
11 HBM Cache
12 Neuromorphic memcapacitive devices
13 Neuromorphic memristor matrix multiplier
14 Neuromorphic asynchronous computing
15 CMOS SRAM CIM
16 CXL optimized DDR5

Advanced Packaging & Heterogeneous Integration

1 LMP solder with polymer
2 Nanostructured thermal interface surface
3 CNT TIM
4 Graphene TIM
5 Graphene interconnects
6 CNT interconnects
7 Rh/Ir interconnect
8 CNT for 3D ICs
9 3D IC MIVs
10 Feveros
11 TSV for 3D IC
12 Hybrid bonding (Cu-Cu)
13 Optical off-chip interconnect
14 Optical on-chip interconnect
15 Optical bus
16 UCIe chiplet standard
17 3D stacked SRAM
18 MIV stacked ReRAM
19 HBM on logic

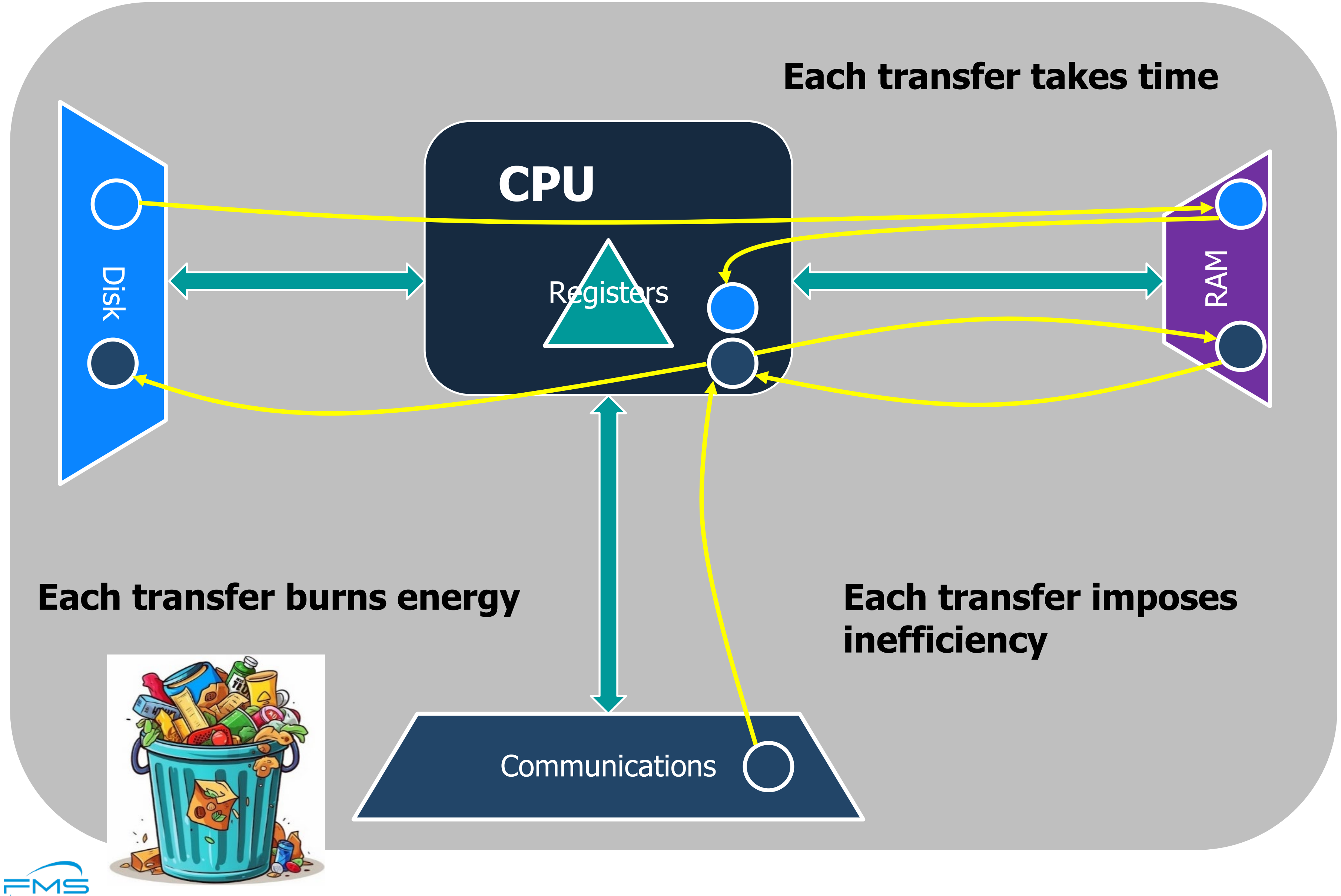
Algorithms and Software

1 Reduced energy for ML algorithms
2 Algorithm-specific energy (tooling)
3 Algorithm-specific energy (benchmarking)
4 Languages, compilers, and runtime systems
5 Communication protocols
6 Homomorphic encryption
7 Software for emerging architectures
8 Computational reliability

Materials and Devices

1 Si-GAA
2 CNT Memory
3 CNTFET (Logic)
4 TFET
5 Spintronic memory
6 FeFET (Flash)
7 Analog devices for neuromorphic computing
8 FeFET (SRAM)
9 Contact & interconnect
10 Novel ILD
11 Spintronic logic
12 2D materials

Simplified but realistic case of program execution and data movement



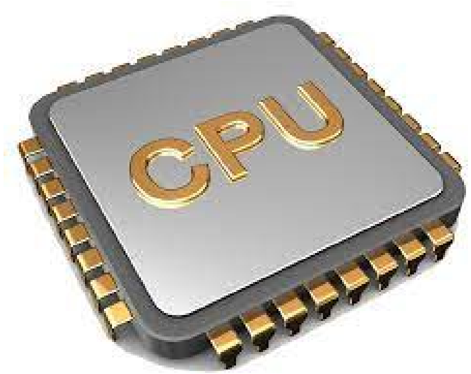
Typical application flow

1. App read from disk through CPU to RAM
2. App read from RAM to CPU for execution
3. Info read from I/O through CPU and written to RAM
4. App reads RAM to process
5. App writes results to disk

Speculation

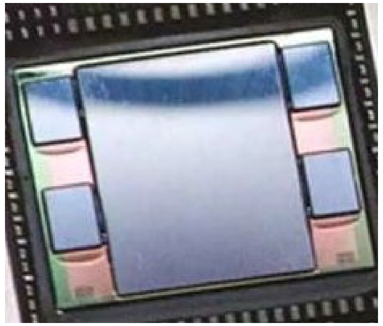
Systems like to do block data moves to "pay" for latency overhead

How often does speculation pay off in terms of operations/watt?



INT8
INT16
FP16
FP32
FP64

CPU registers have an intrinsic waste with various size data types



CPUs have all added caches for recently accessed data

Industry standard is 64 bytes per cache line

64 Byte Cache Line

If an application needs a yes or no answer (1 bit)

But accesses a cache line (64 bytes)

Discrepancy between cache line size and data item size creates significant wasted data access

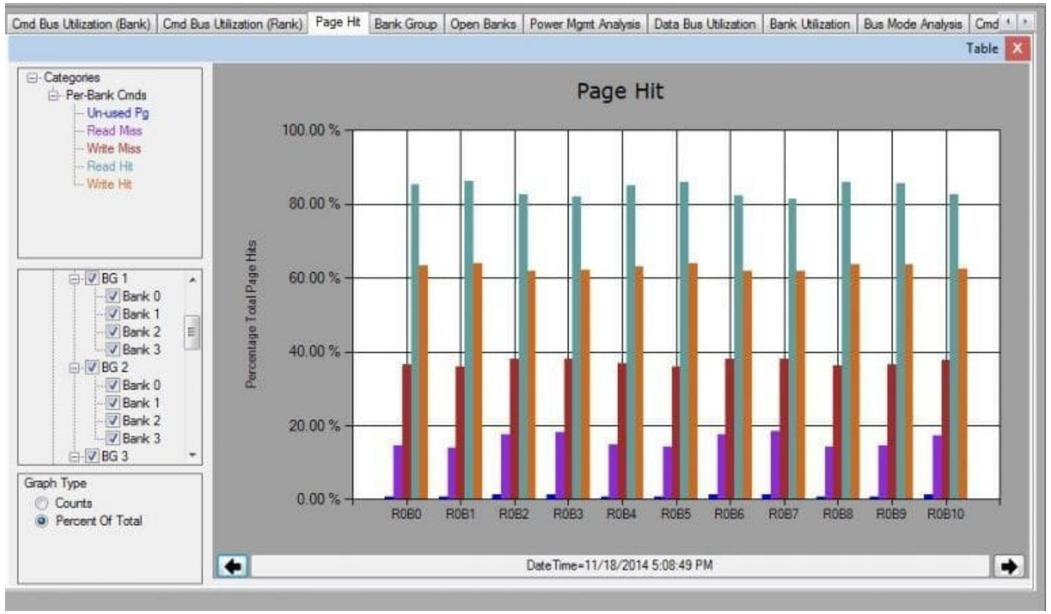
Waste = 99.8%

L1: 96% hit rate, 1 cycle access
L2: 95% hit rate, 25 cycles access
L3: 98% hit rate, 80 cycles access

The good news: near-CPU caches do have high hit rates (reduces waste from unnecessary accesses)

By the time an access gets to the local DRAM, though, hit rates start to drop dramatically

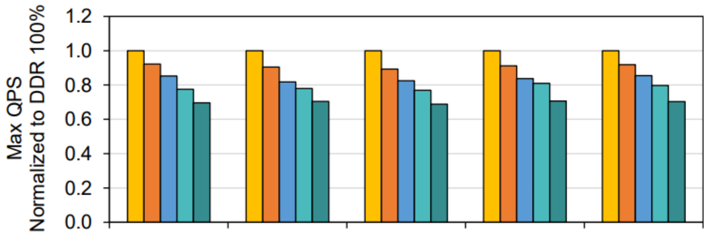
Read hit ~82%
Write hit ~62%



A question I have posed that CPU guys refuse to answer:

How much performance gain are we getting for each watt expended?

ESPECIALLY when it comes to speculative operations



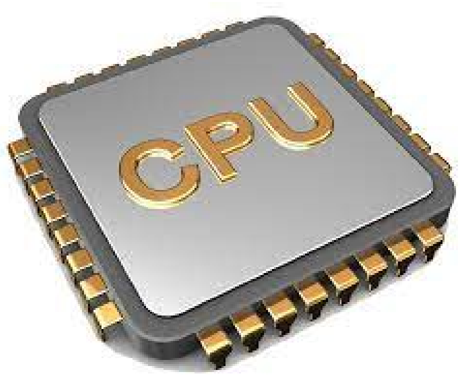
Access to remote memory drops even further, especially with increased thread count

Hit rate ~65%

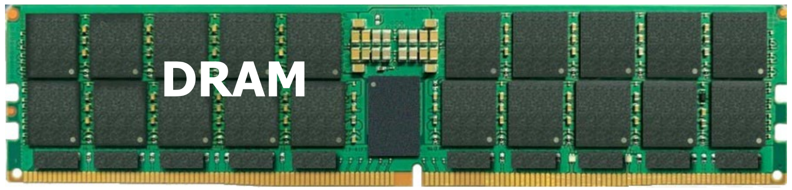
…and this is before memory pooling…

https://www.futureplus.com/blog/critical-memory-performance-metrics-for-ddr4-systems-page-hit-analysis

https://arxiv.org/pdf/2303.15375#:~:text=Meanwhile%2C%20as%20the%20block%20size%20increases%20beyond,latency%20begins%20to%20dominate%20the%20p99%20latency.

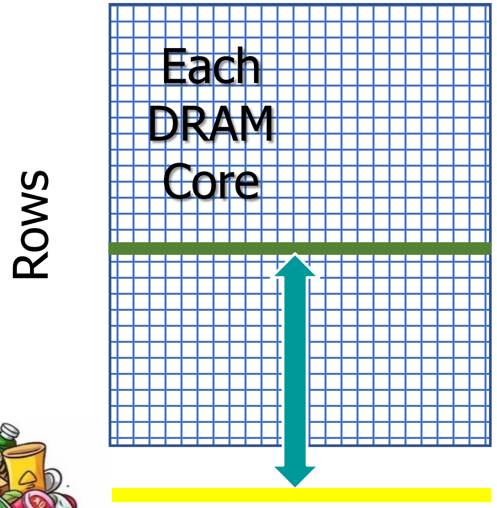


64 byte
cache line



10KB block
X 2

Columns



Page Buffer

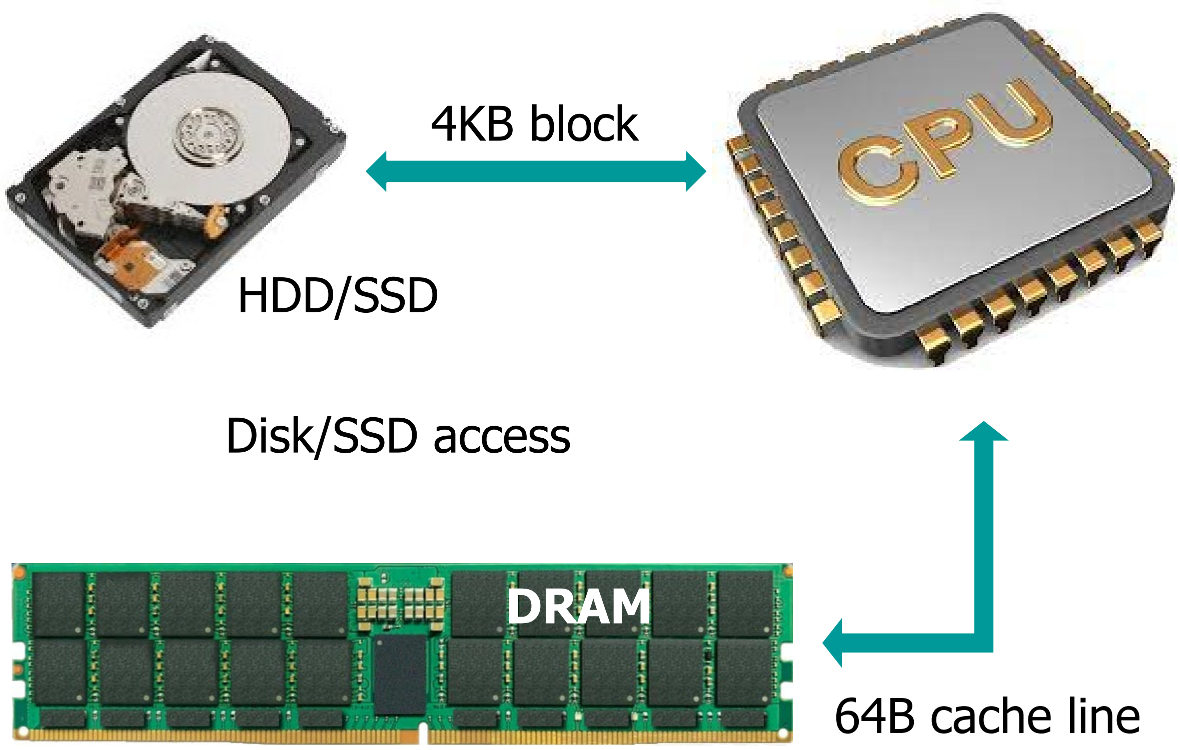
RAMs are grouped in 10s to form a "rank"

Each RAM has a 1KB page buffer size (access granularity)

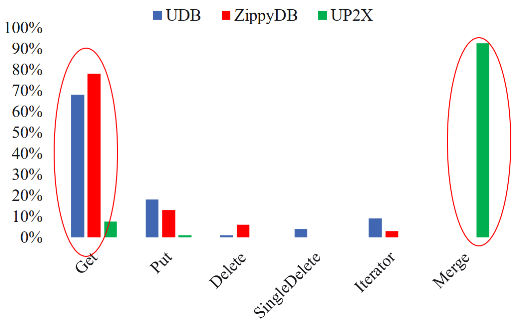
Activations are destructive and data rewrite is needed

Therefore, every data access requires 20KB of data movement

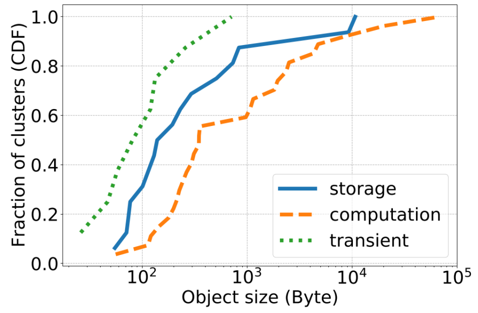
Waste = 99.7%



Facebook RocksDB



X (Twitter) Twemcache



The average key size (AVG-K), the standard deviation of key size (SD-K), the average value size (AVG-V), and the standard deviation of value size (SD-V) of UDB, ZippyDB, and UP2X (in bytes)

	AVG-K	SD-K	AVG-V	SD-V
UDB	27.1	2.6	126.7	22.1
ZippyDB	47.9	3.7	42.9	26.1
UP2X	10.45	1.4	46.8	11.6

Typical disk block transfer size is 4KB

Average number of bytes actually used is 100

Waste = 97.5%

This is Best Case… *even worse if the block is cached*

Reducing Power

1. Let's get smarter about speculation accesses and do a TCO analysis on each
2. Consider the number of data hops implied by each access
3. Move the processing to the data when possible, not the data to the processing
4. Let's work on protocols that minimize unnecessary data movement

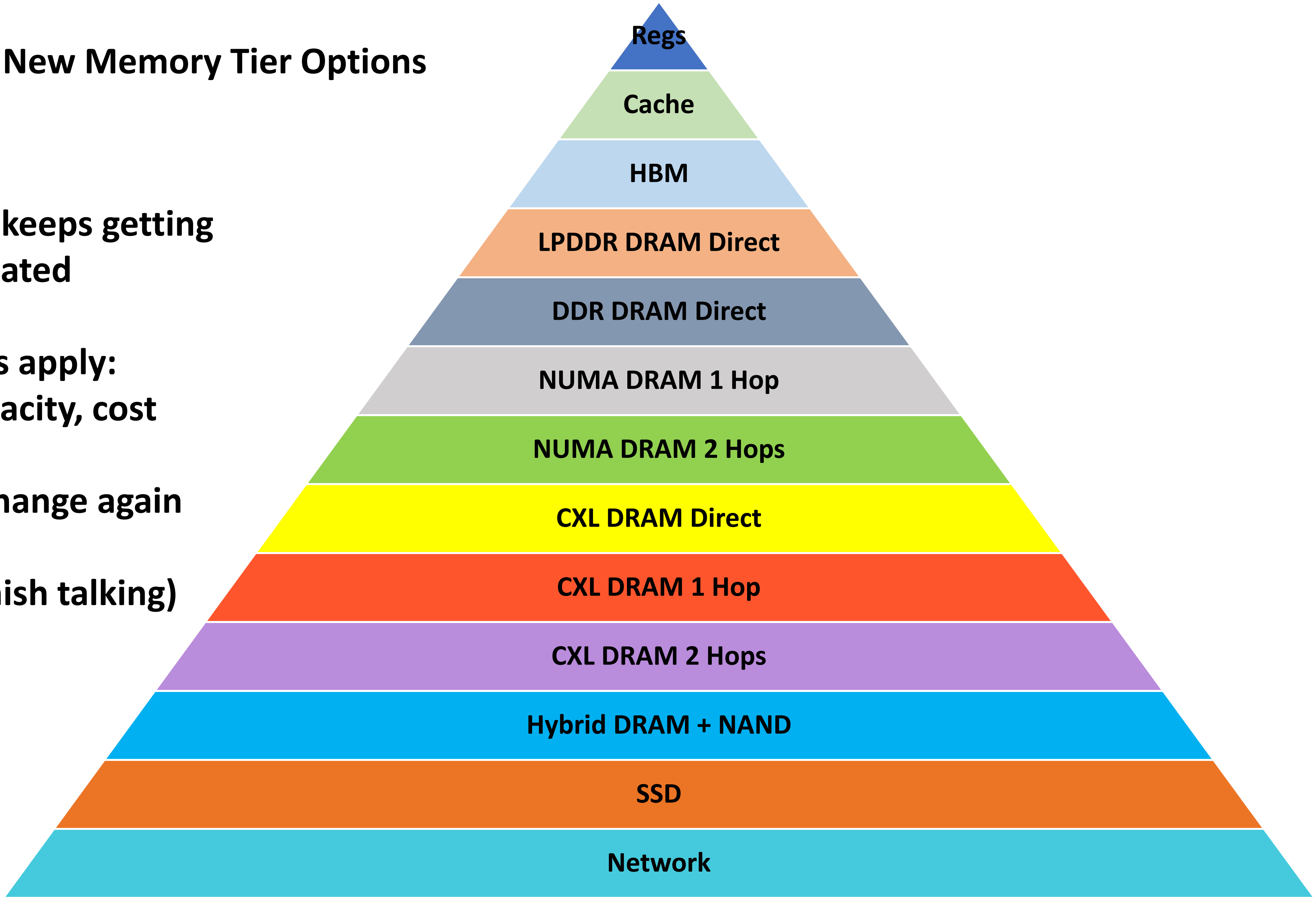
Introducing New Memory Tier Options

The resource tier map keeps getting more complicated

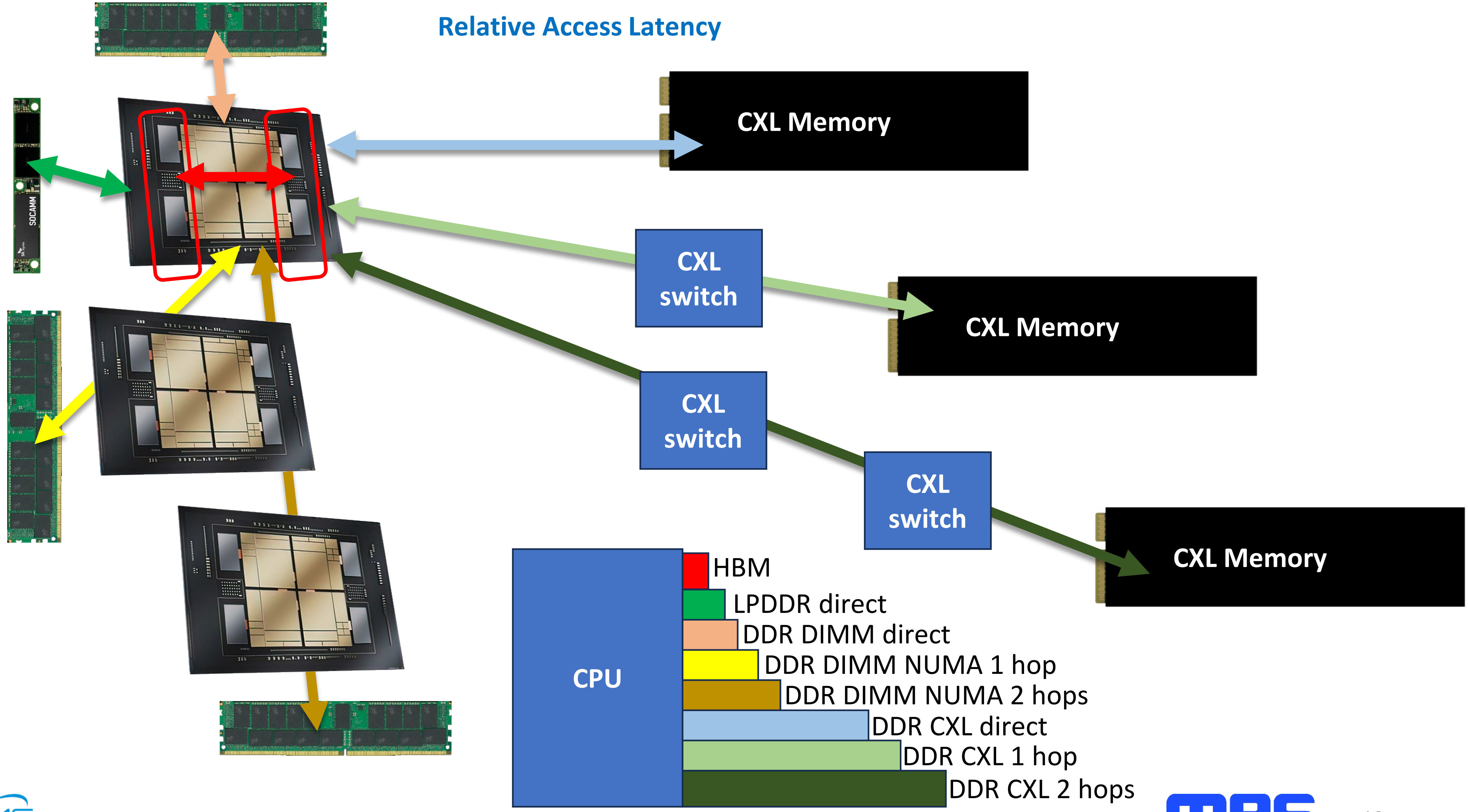
The same factors apply:
speed, latency, capacity, cost

Don't blink. It will change again

(Possibly before I finish talking)



Relative Access Latency

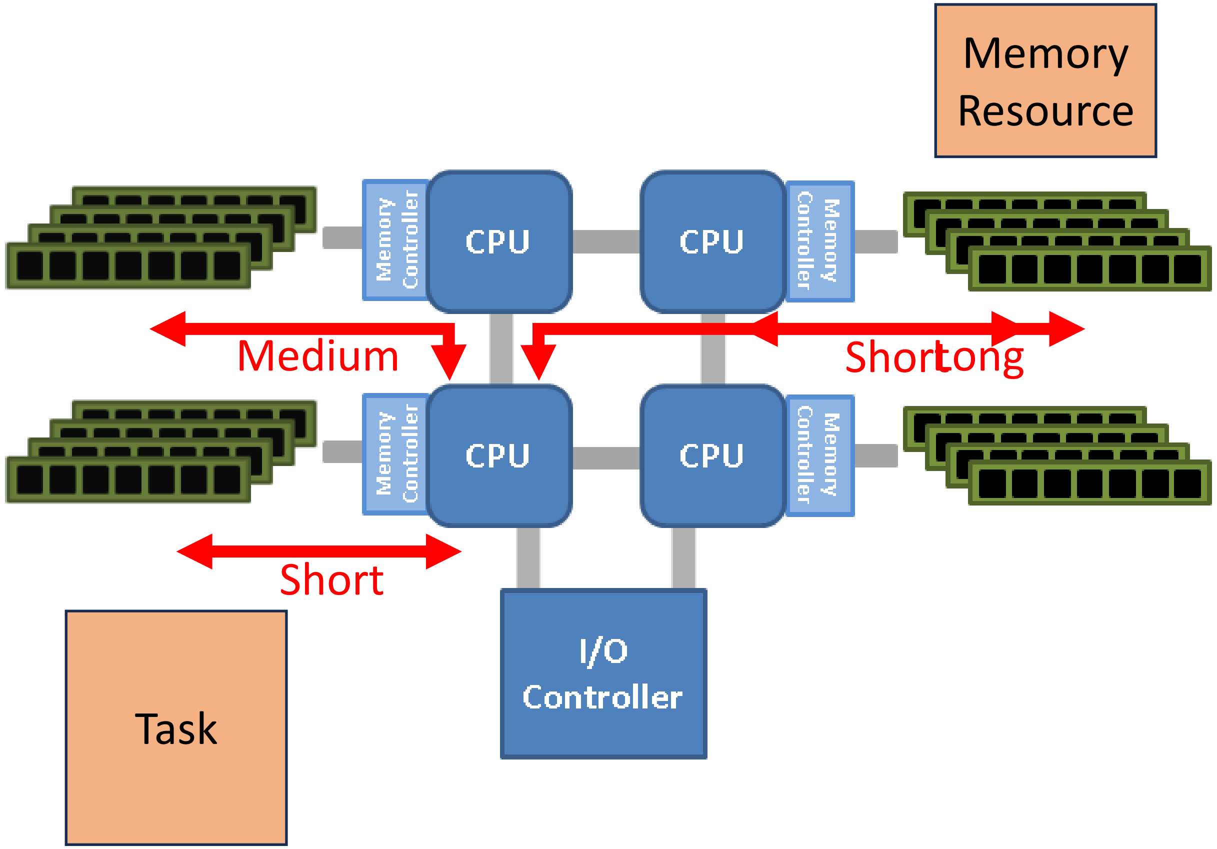


NUMA doesn’t have to be just about sharing memory

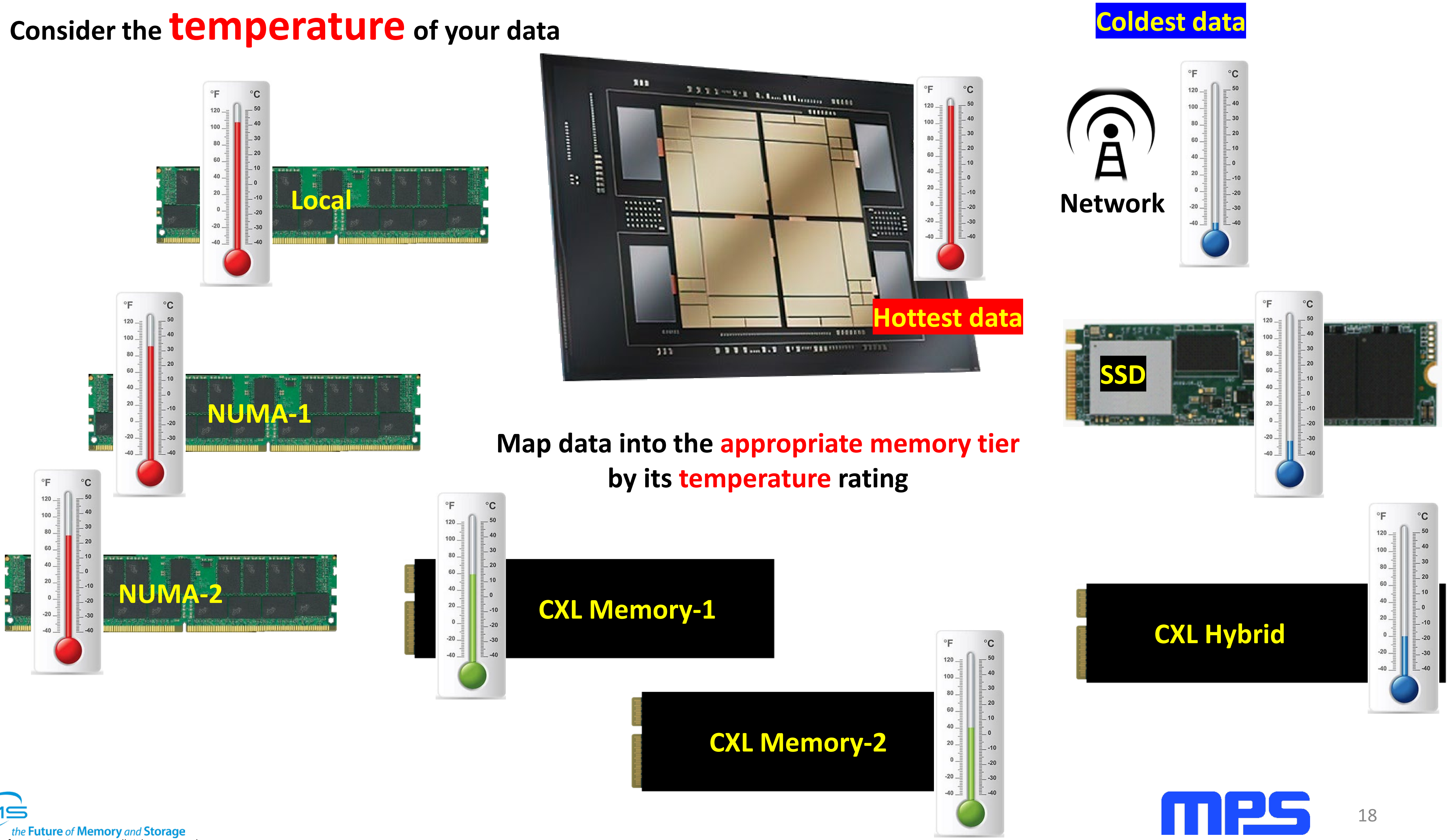
Job distribution can potentially save power and improve performance

Rather than grab a memory resource over NUMA…

…Move the task to the memory

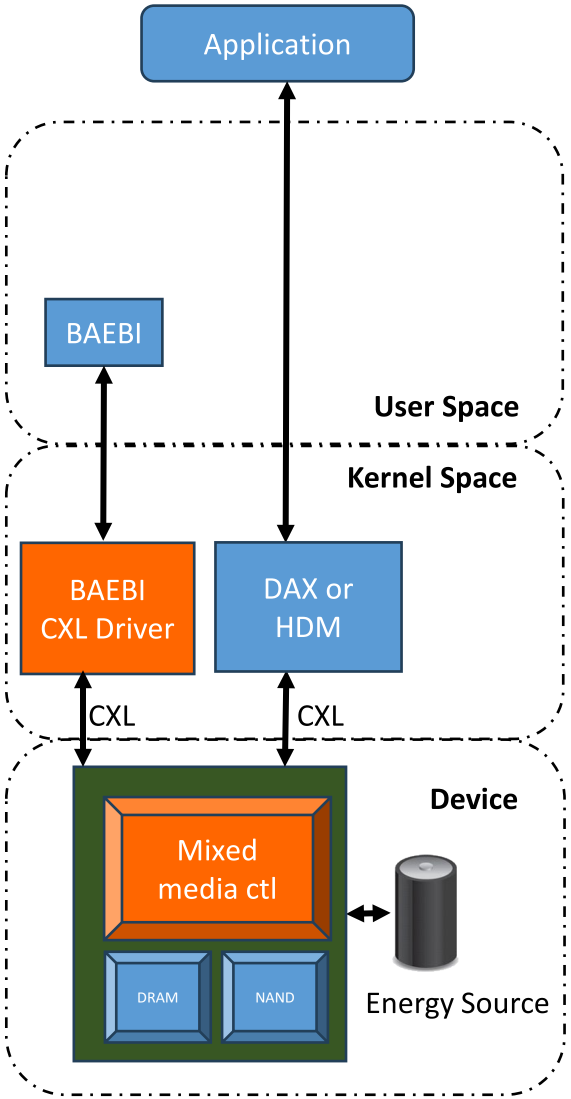


Consider the temperature of your data

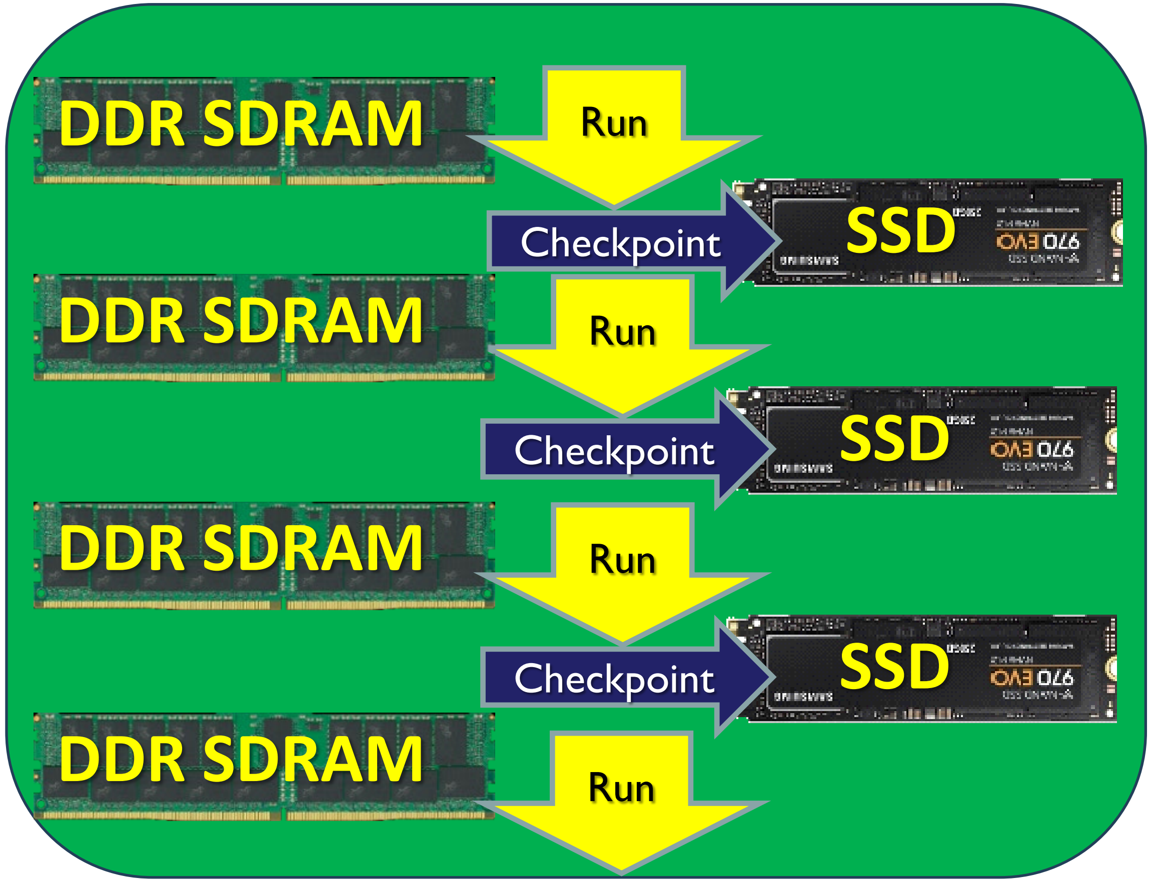


Map data into the appropriate memory tier
by its temperature rating

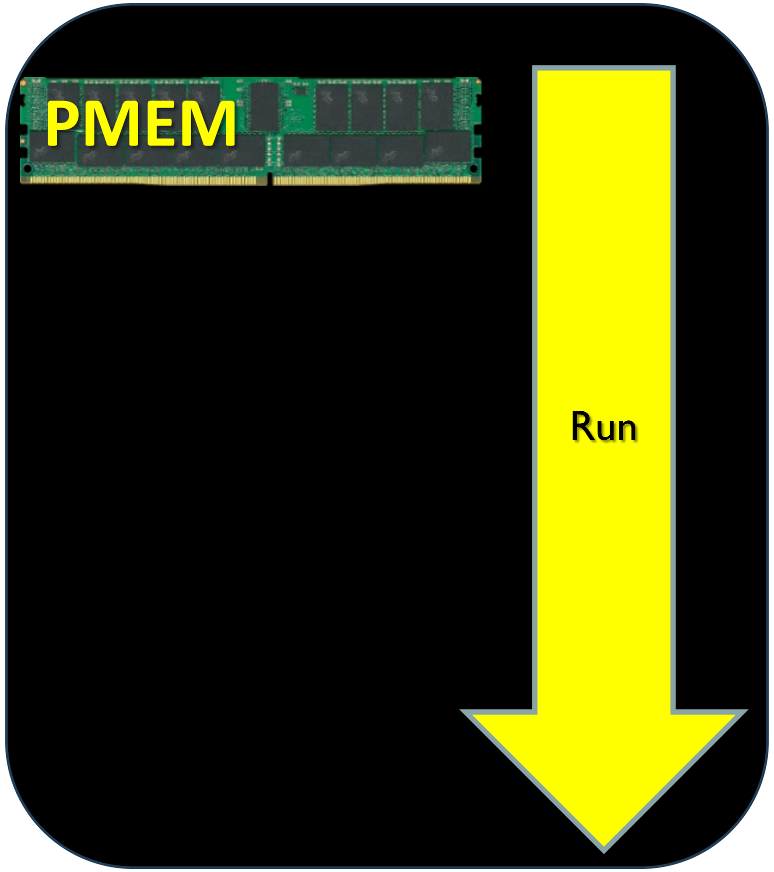
Persistent memory is not just about data integrity



Applications are forced to checkpoint contents periodically because of volatile DRAM

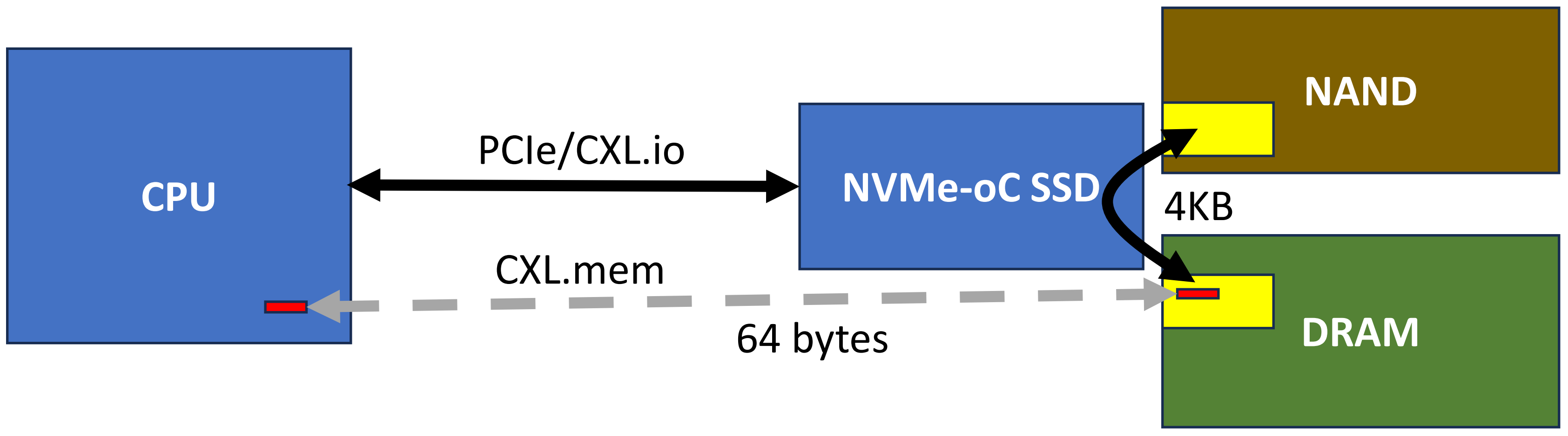


Checkpointing consumes ~8% of system throughput and power on average



Persistent memory can save power

NVMe Over CXL: Only grab the FLITs you need



NVMe is just a cache protocol between NAND and DRAM

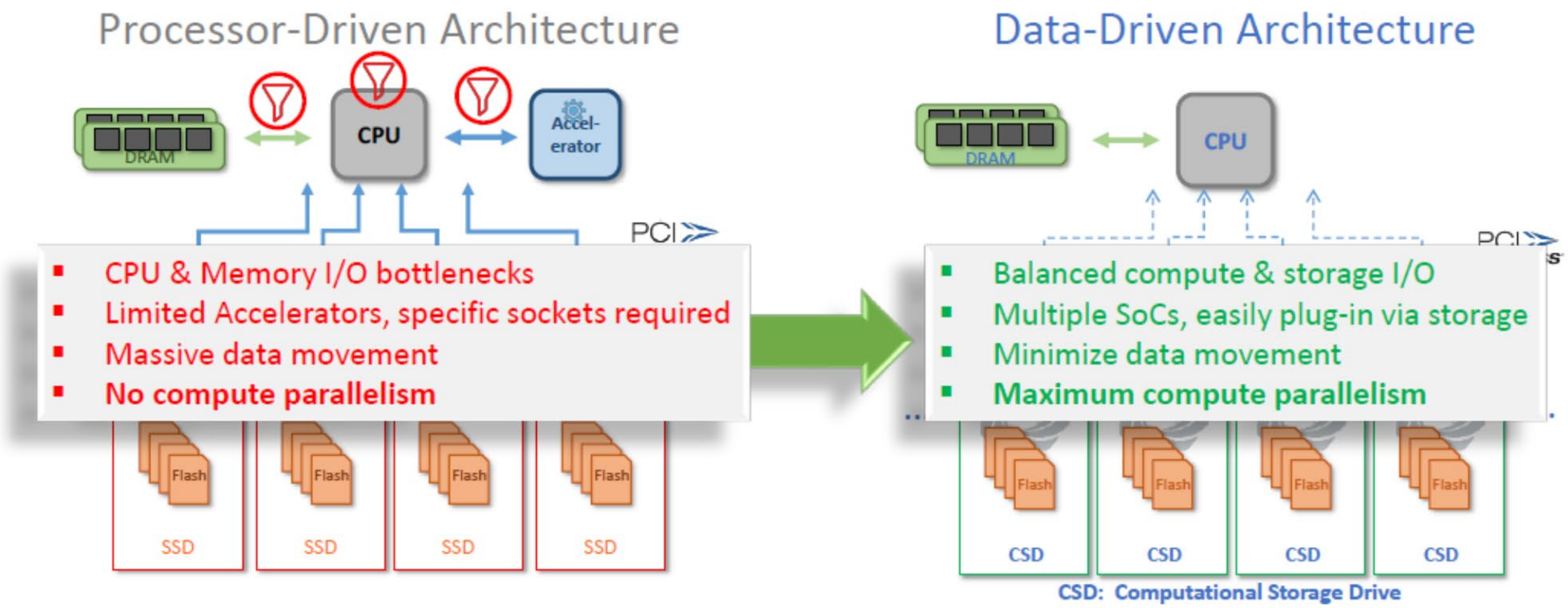
NVMe-oC places the controller memory buffer (CMB) in CXL space (HDM)

Processor grabs only the FLITs needed using CXL.mem

The rest of the CMB data (on average, 97%) remains where it is

This cache management scheme is expanded to create Virtual HDM

Computational storage –
Another way to move the processing to the data



Significant challenges:
- No vendor interoperability
- May not accelerate versus CPU consistently
- Programming complexity

Potential savings:
- Power reduction
- Ease of checkpointing
- Reduce CPU workload

Summary

We rock at moving data around!

We are TERRIBLE at using that data!

We are burning down the world!

We can do better!

Thank you for your time

Any questions?

Bill Gervasi, Principal Memory Solutions Architect

Monolithic Power Systems

bill.gervasi@monolithicpower.com