Flash Memory Summit

Not a magic wand!

Performance Considerations for CXL Attached Memory

Nishant Lodha, Marvell

Disclaimer

This presentation contains forward-looking statements within the meaning of U.S. federal securities laws that involve risks and uncertainties. Forward-looking statements include, without limitation, any statement that may predict, forecast, indicate or imply future events or achievements. Actual events or results may differ materially from those contemplated in this presentation. Forward-looking statements are only predictions and are subject to risks, uncertainties and assumptions that are difficult to predict, including those described in the “Risk Factors” section of our Annual Reports on Form 10-K, Quarterly Reports on Form 10-Q and other documents filed by us from time to time with the SEC. Forward-looking statements speak only as of the date they are made. Readers are cautioned not to put undue reliance on forward-looking statements, and no person assumes any obligation to update or revise any such forward-looking statements, whether as a result of new information, future events or otherwise.

A room called “memory”

#1
Spending category for Cloud.
50% of server cost of memory.

>50%
Server cost of memory

>50%
Of allocated application memory idle in any 2-minute interval (Meta)

70%
Dual or more socketed servers in data centers

25%
Stranded memory (Azure)

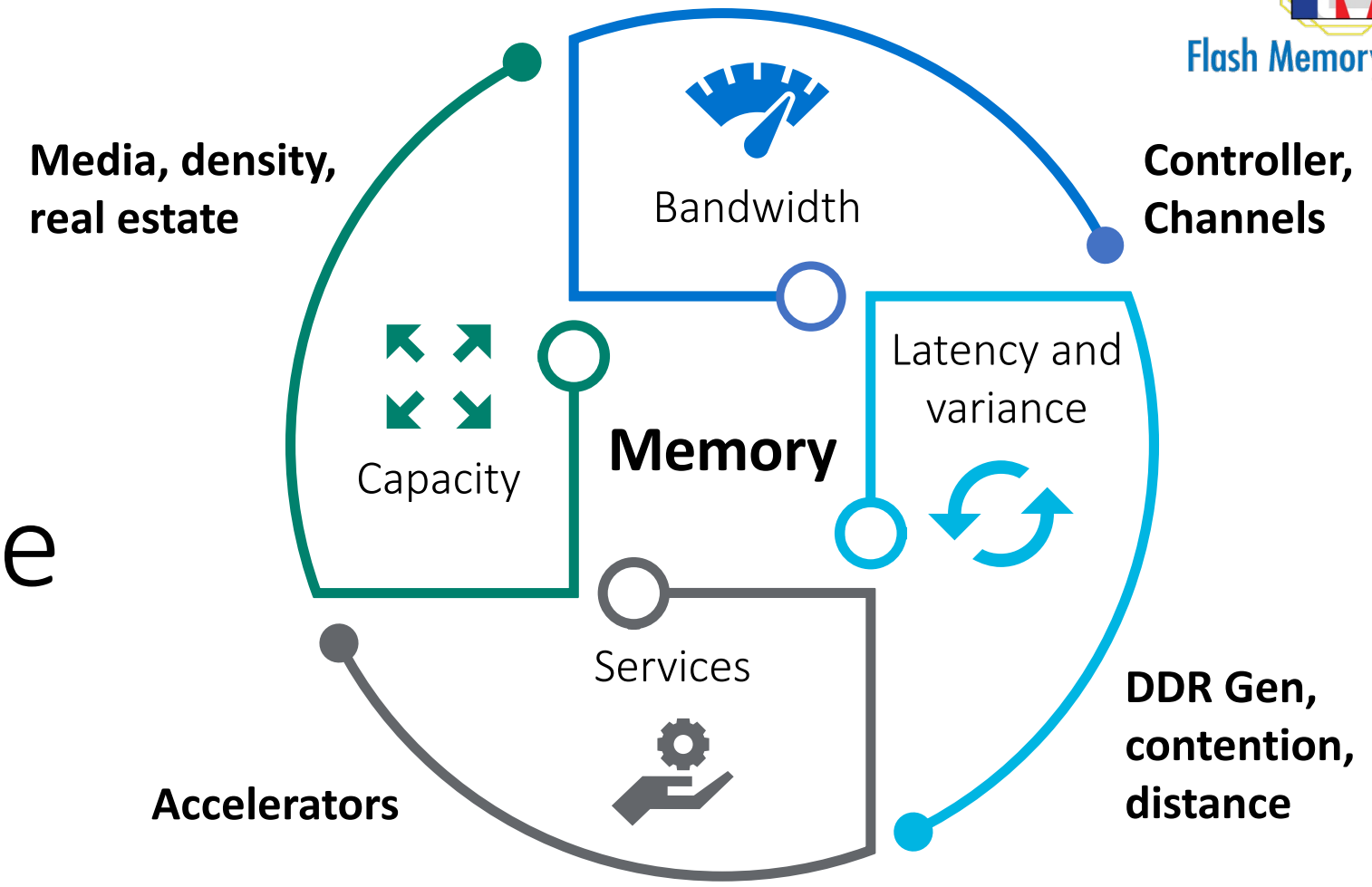
0
Backward compatibility (of DDR-x across processor generations)

Significant **Unused** life of DDR

1
Viable standard | CXL

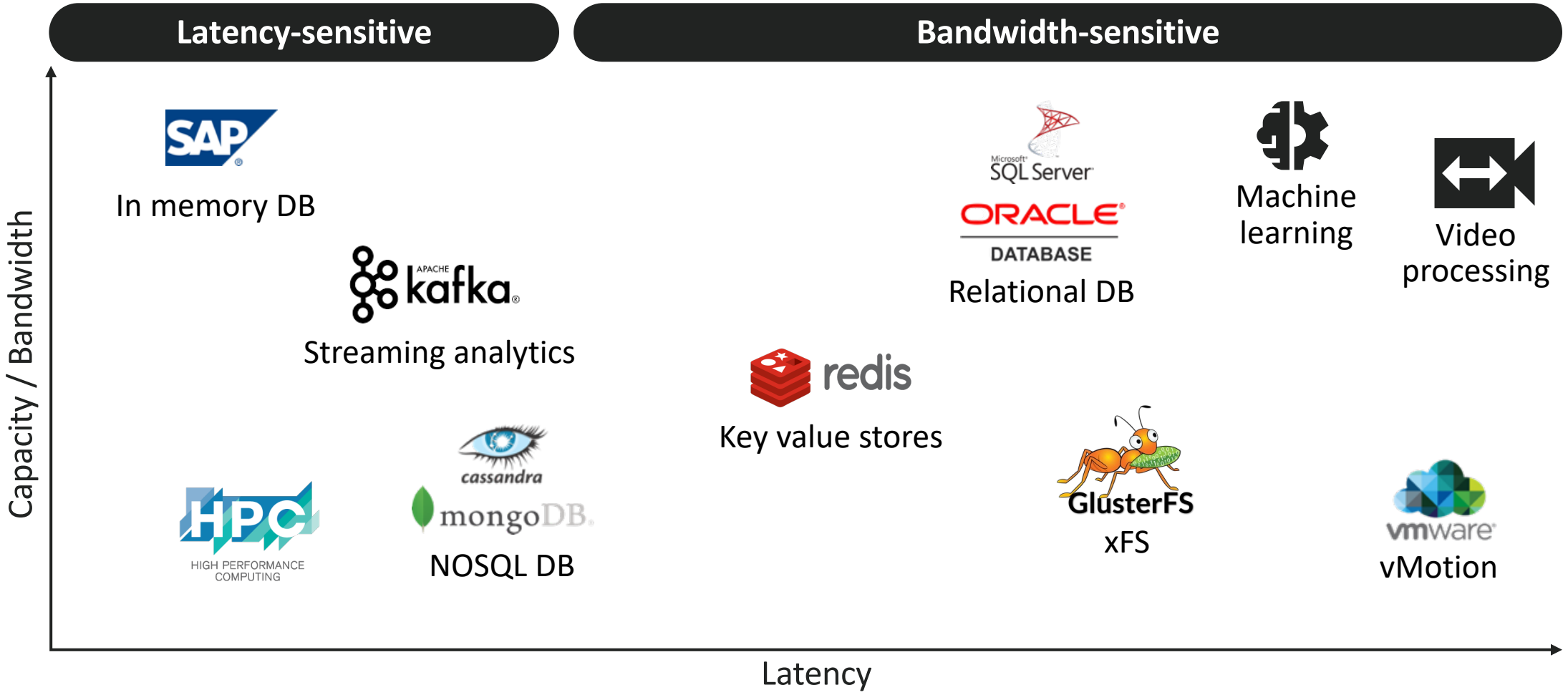
Various sources and published papers

The memory performance puzzle



The “why” for memory disaggregation will determine weight

Workload and their memory demands

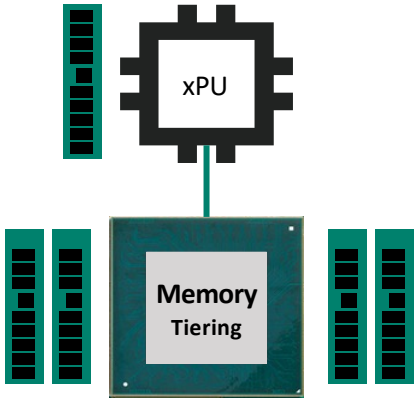


Logos are indicative of workload characteristics only. Actual position on chart is highly subjective to business application, time of day etc.

Performance mapping of CXL mem use cases

Flash Memory Summit

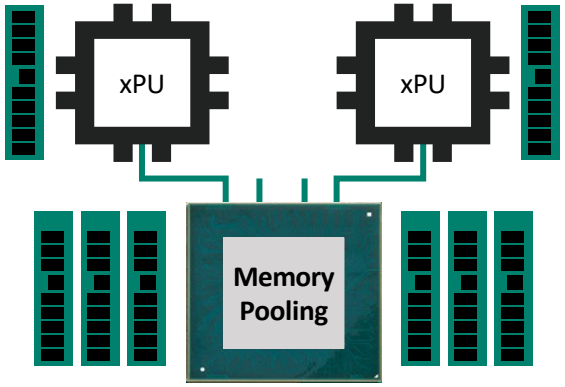
Memory tiering



Bandwidth ↑
Latency ↓
Capacity ↑ ↑
Utilization ↓

RoI for capacity sensitive applications with memory re-use centric

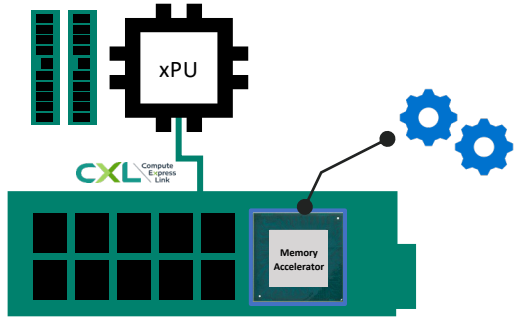
Memory pooling



Bandwidth ↑
Latency ↑
Capacity ↑
Utilization ↑ ↑

RoI @ <16sockets, >25% CXL memory

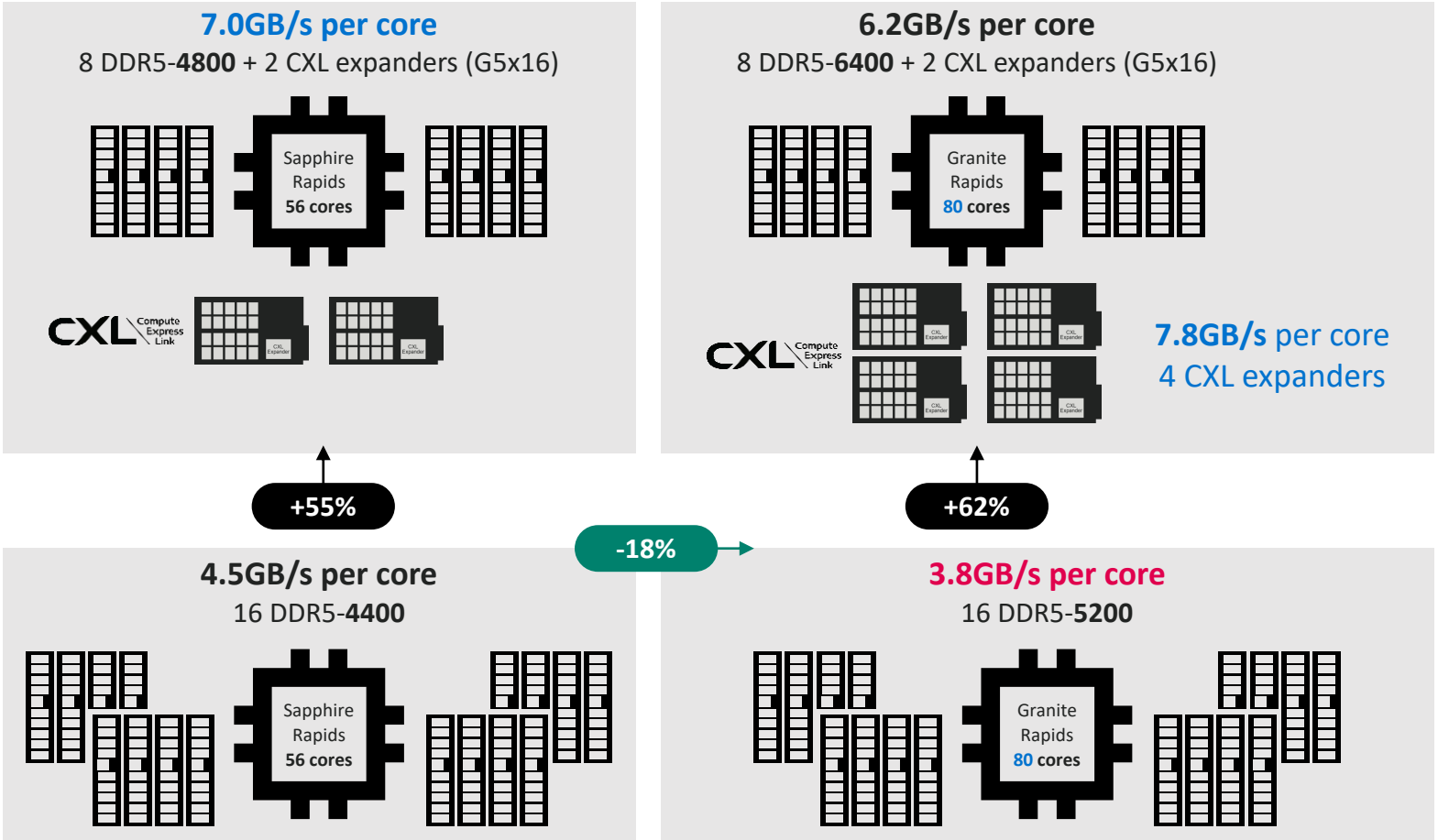
Near memory compute



Bandwidth ↑
Latency ↓ ↓
Capacity ↑
Utilization ↓

Beyond just CXL Emerging use cases

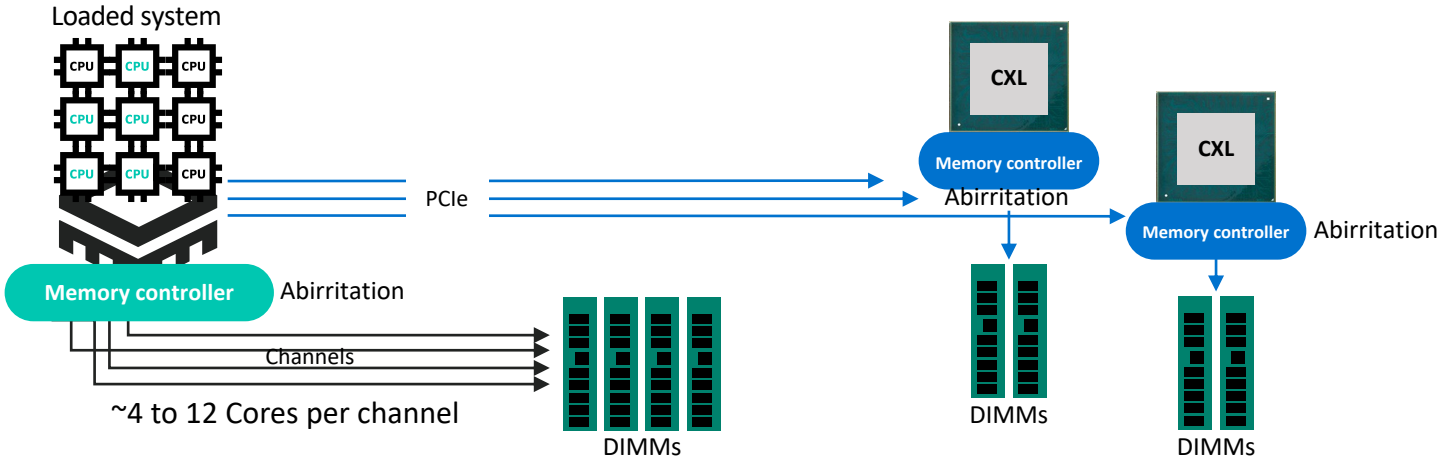
CXL improves bandwidth per core

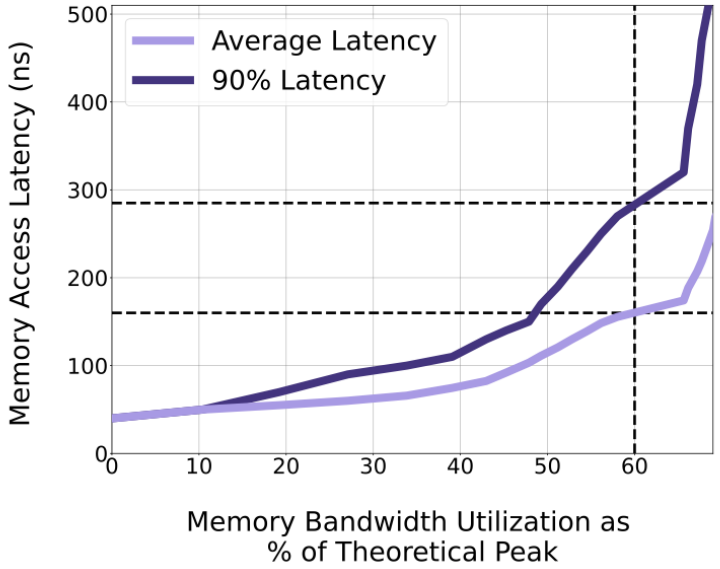


- DDR5 → 1DPC
- 4x more: PCIe 5.0 vs. DDR5
- DDR total bandwidth vs. PCIe bidirectional
- Capacity and throughput-oriented workloads

For latency, “average” is the enemy!

- The concept of “average” and “unloaded” latency are mythical
- Queuing delays due to abirritation across channels dominates memory access latency
- “Loaded” latency characterization factors in real world CPU core and memory bandwidth utilization along with application read write patterns
- CXL latency adder less significant in real life scenarios
- CXL Memory Controller parallelism with CXL could improve relative utilization.





Source: https://arxiv.org/abs/2305.05033

CXL scalable memory controllers enable better parallelism

Going down the memory lane (latency adders)

CPU	CXL Port	Retimers	CXL Switch	MHD	Services	DRAM	
~40ns						45-70ns	~Local
~40ns	~30ns					45-70ns	Tiering
~40ns	~30ns			~45ns	~10ns	45-70ns	Pooling
~40ns	~30ns	~25ns	~85ns	~45ns	~10ns	45-70ns	Pooling++

Latency metrics are indicative and based on my estimates and do not indicate any specific product capability. Actual values are highly dependent on real world conditions and actual products deployed

Many factors: Multiplexing multiple memory devices, enforcing coherence, memory services, software overhead, switching, cabling, re-timers

No magic wand!

We need to answer the “why” and “where” first!

Thanks!