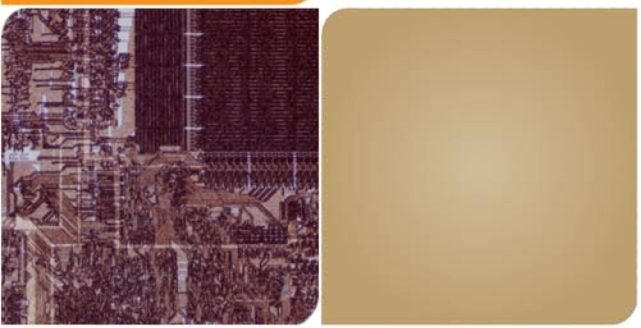
Performance Implications of NoCs on 3D-Stacked Memories: Insights from the Hybrid Memory Cube (HMC)

Ramyad Hadidi, Bahar Asgari, Jeffrey Young, Burhan Ahmad Mudassar, Kartikay Garg, Tushar Krishna, and Hyesoon Kim



Georgia Tech

comparch

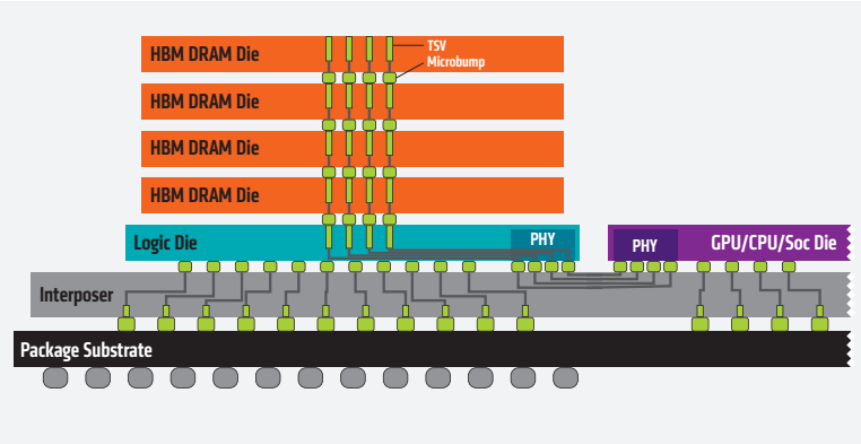
SYNERGY

CASL
Computer Architecture and Systems Laboratory

Introduction to HMC

Hybrid Memory Cube (HMC) vs High-Bandwidth Memory (HBM)

- HMC: Serial, packet-based interface
- HBM: Wide bus, standard DRAM protocol
  - Found in high-end GPUs and Intel's Knight's Landing



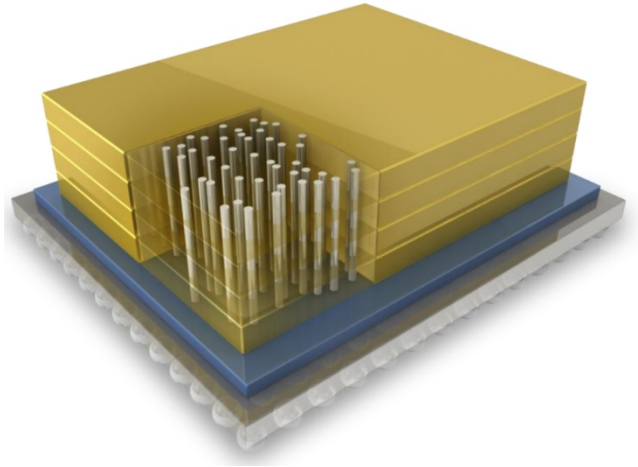


Illustration credits: AMD and Micron

Why is HMC Interesting?

-Serialized, high-speed link addresses pin limitation issues with DRAM and HBM

-Abstracted packet interface provides opportunities for novel memories and addressing opportunities

- Can be used with DRAM, PCM, STT-RAM, NVM, etc.

-Memory controller sits on top of a “routing” layer

- Allows for more interesting connections between processors and memory elements
- This study addresses the impacts of the network on chip (NOC) for architects/application developers

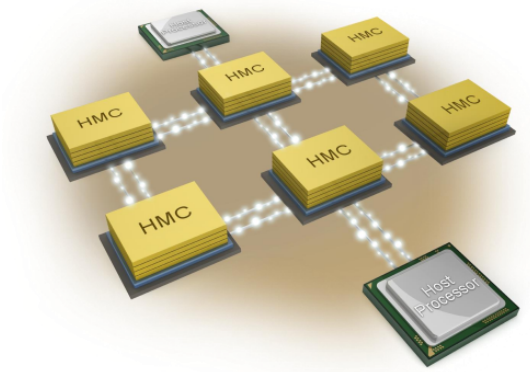
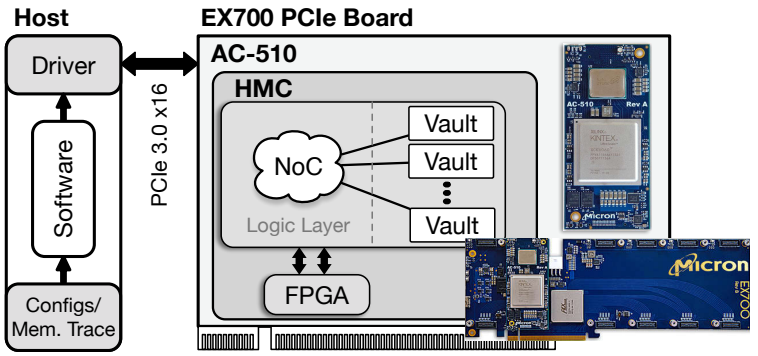


Illustration credits: Micron

This Study's Contributions

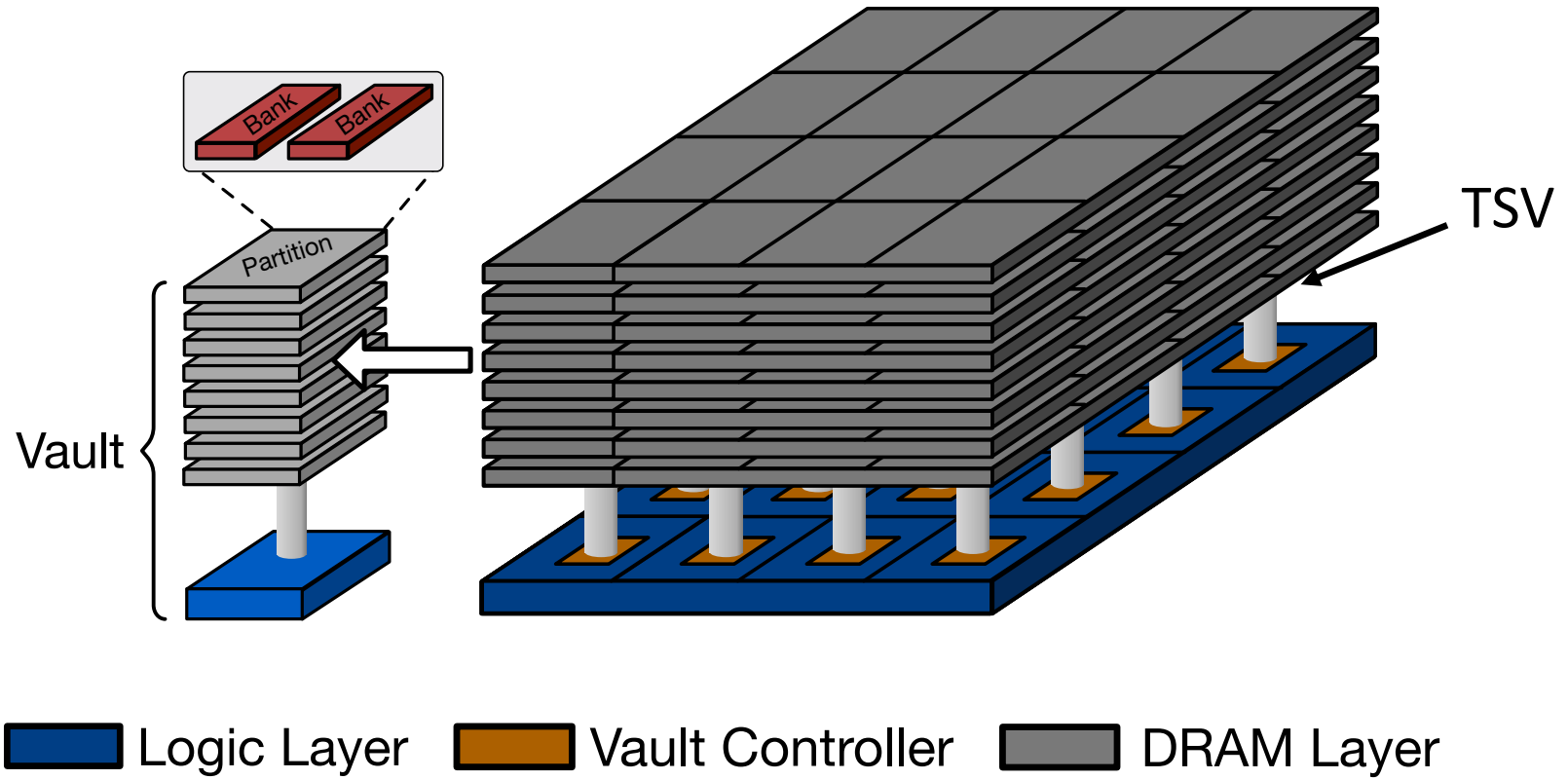
We examine the NoC of the HMC using an FPGA-based prototype to answer the following:

1) How does the NoC behave under low- and high-load conditions?
2) Can we relate QoS concepts to 3D stacked memories?
3) How does the NoC affect latency within the HMC?
4) What potential bottlenecks are there and how can we avoid them?



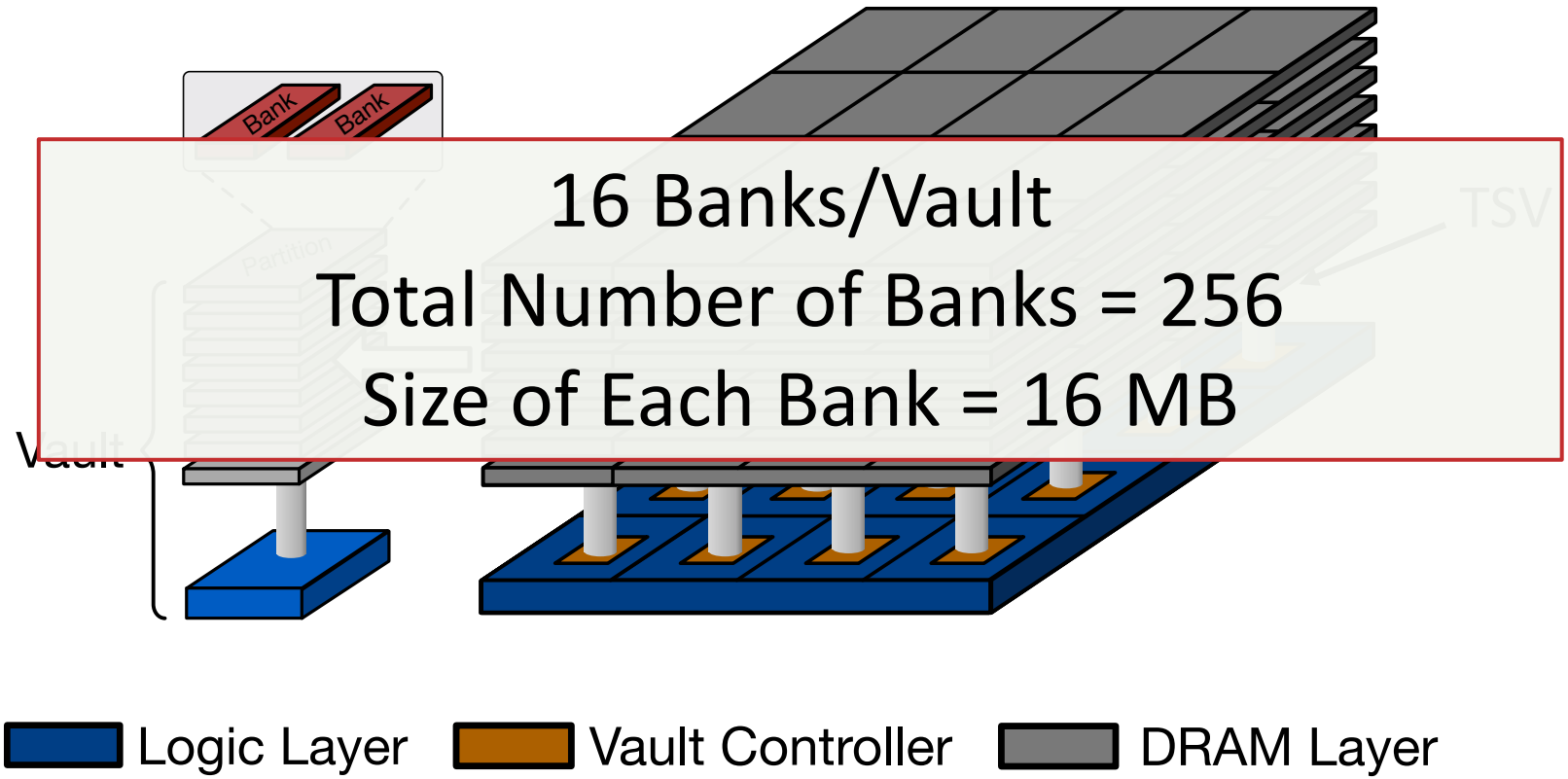
Hybrid Memory Cube (HMC)

HMC 1.1 (Gen2): 4GB size



Hybrid Memory Cube (HMC)

HMC 1.1 (Gen2): 4GB size

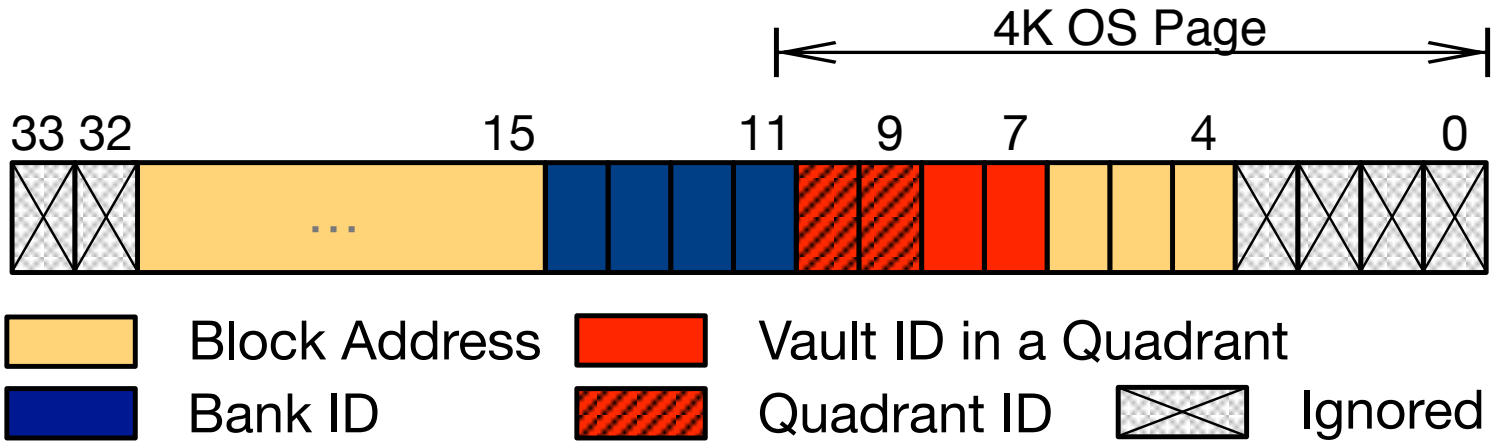


HMC Memory Addressing

Closed-page policy Page Size = 256 B

Low-order-interleaving address mapping policy

34-bit address field:



HMC Communication I

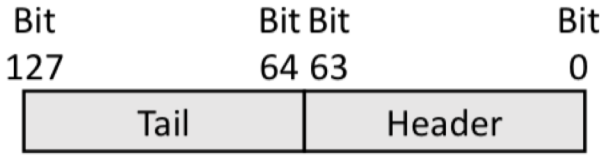
Follows a serialized **packet-switched** protocol

Partitioned into 16-byte *flit*

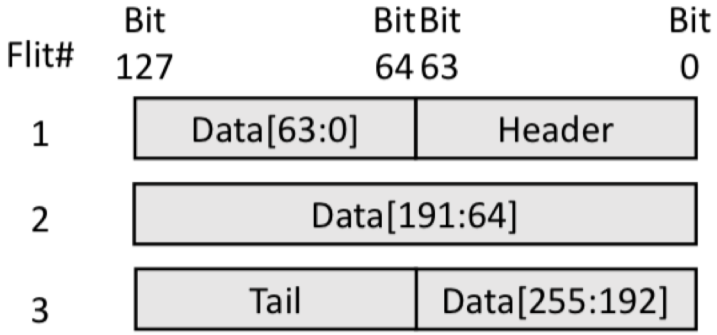
Each transfer incurs 1 flit of overhead

Type	Request		Response	
	Read	Write	Read	Write
Data Size	Empty	1∼8 Flits	1∼8 Flits	Empty
Overhead	1 Flit	1 Flit	1 Flit	1 Flit
Total Size	1 Flit	2∼9 Flits	2∼9 Flits	1 Flit

HMC Communication II

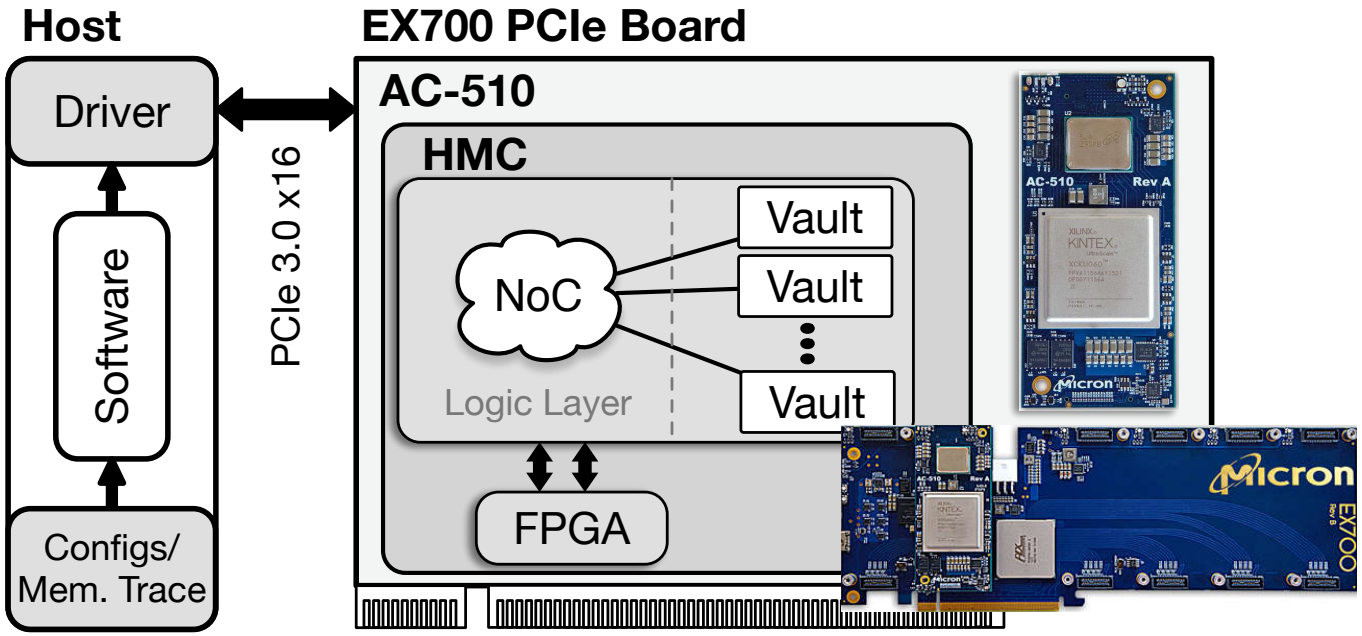


(a)
Flow Control



(b)
Request/Response

Our HMC Test Infrastructure

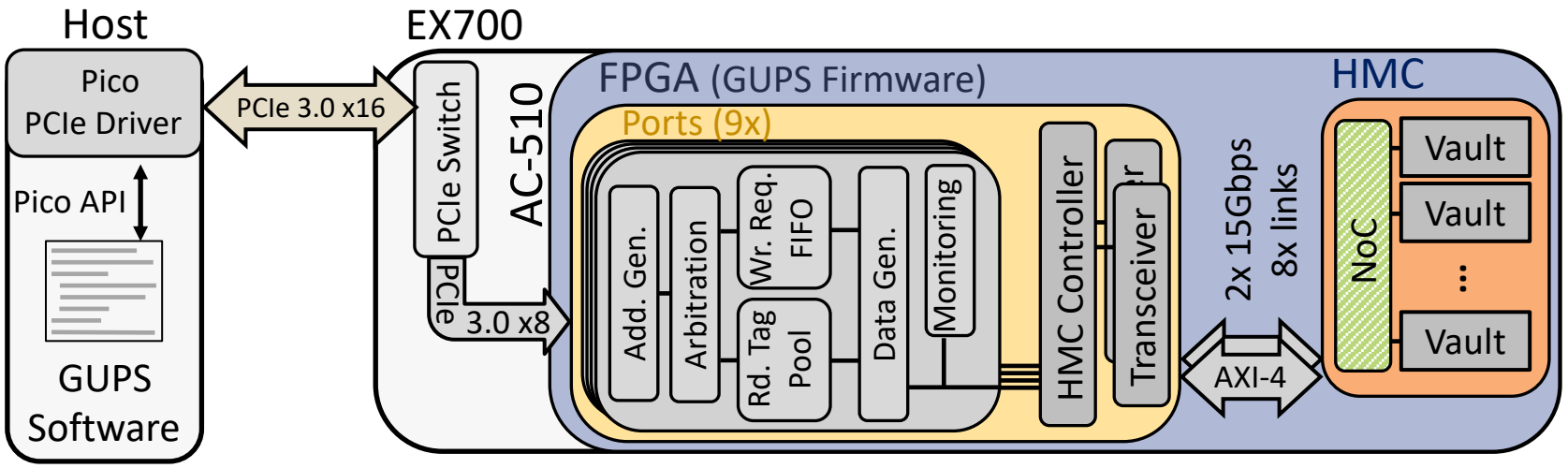


-Micron's AC-510 module contains a Xilinx Kintex FPGA and HMC 1.1 4 GB part

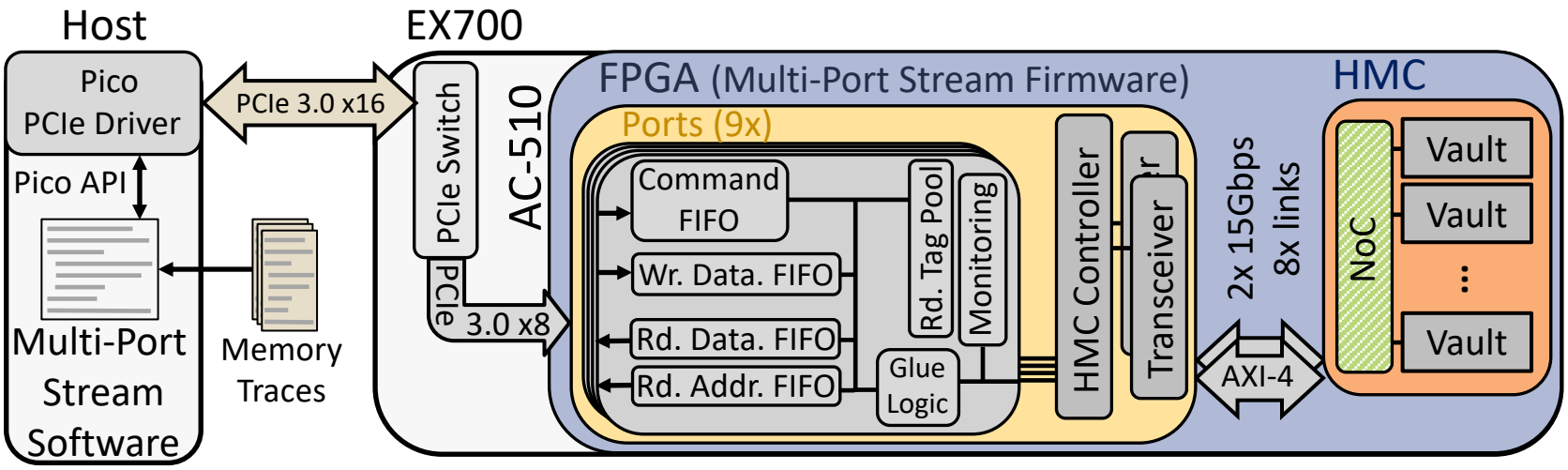
-2 half-width links for a total of 60 GB/s of bandwidth

-Host SW communicates over PCIe to FPGA-based queues

Methodology (GUPS)



Methodology (multi-port stream)



Experiments

[1] High-Contention Latency Analysis **(GUPS design)**

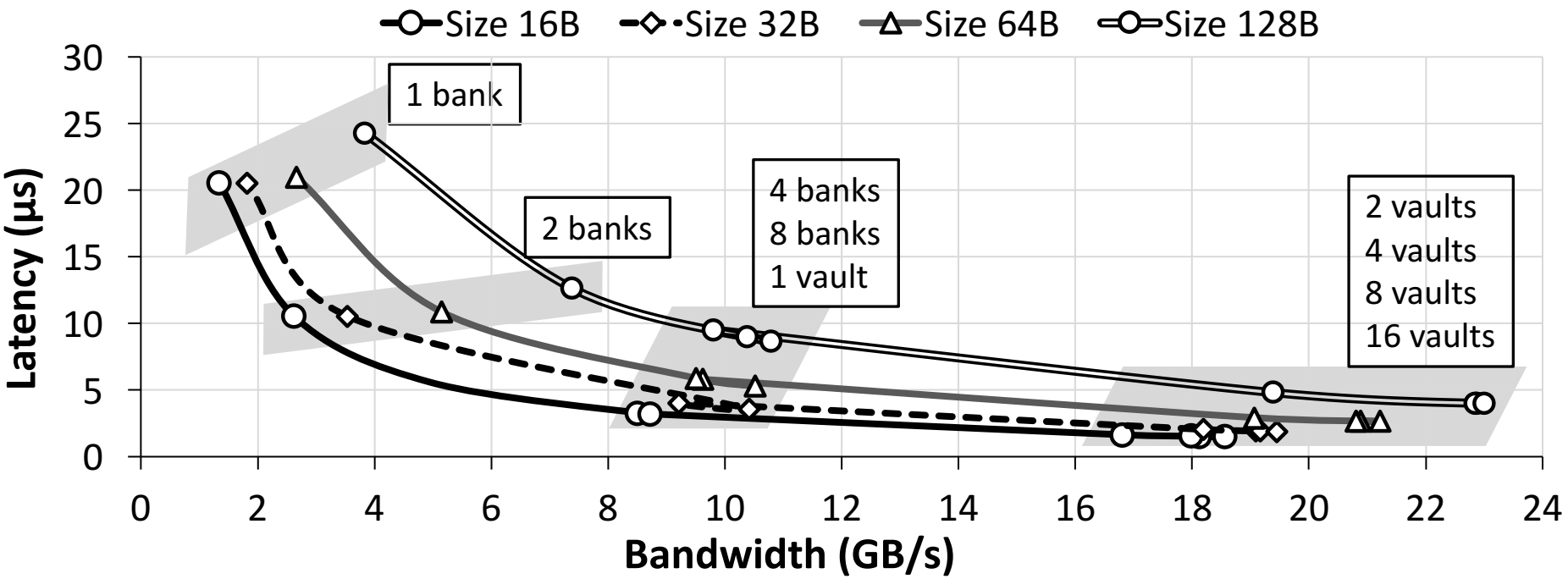
[2] Low-Contention Latency Analysis **(Multi-port stream)**

[3] Quality of Service Analysis **(Multi-port)**

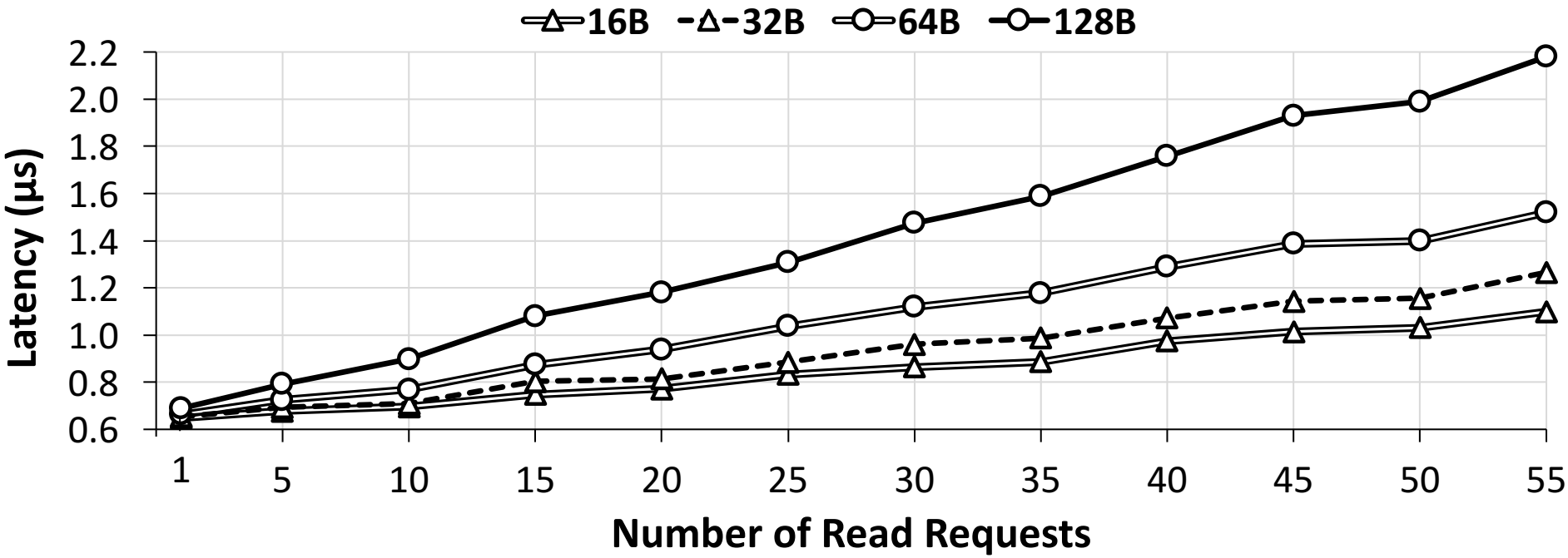
[4] High-Contention Latency Histograms Per Vault **(Multi-port)**

[5] Requested and Response Bandwidth Analysis **(GUPS)**

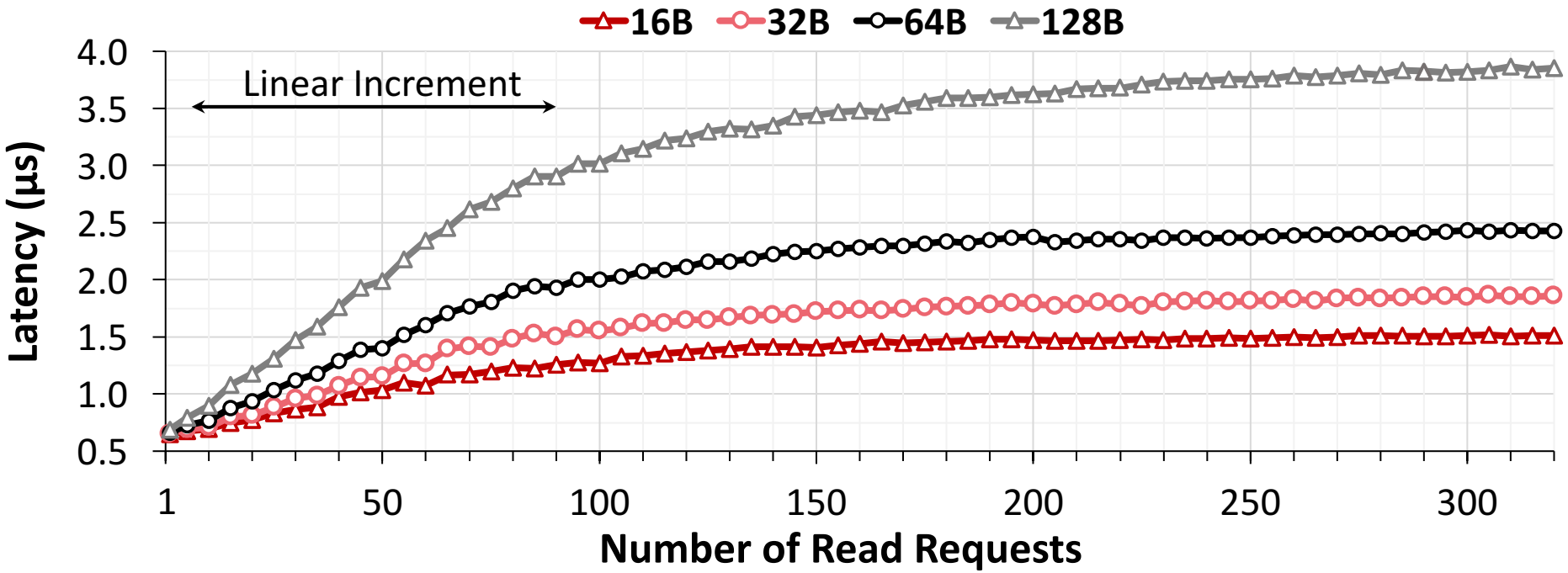
[1] Read-only Latency vs. Bandwidth



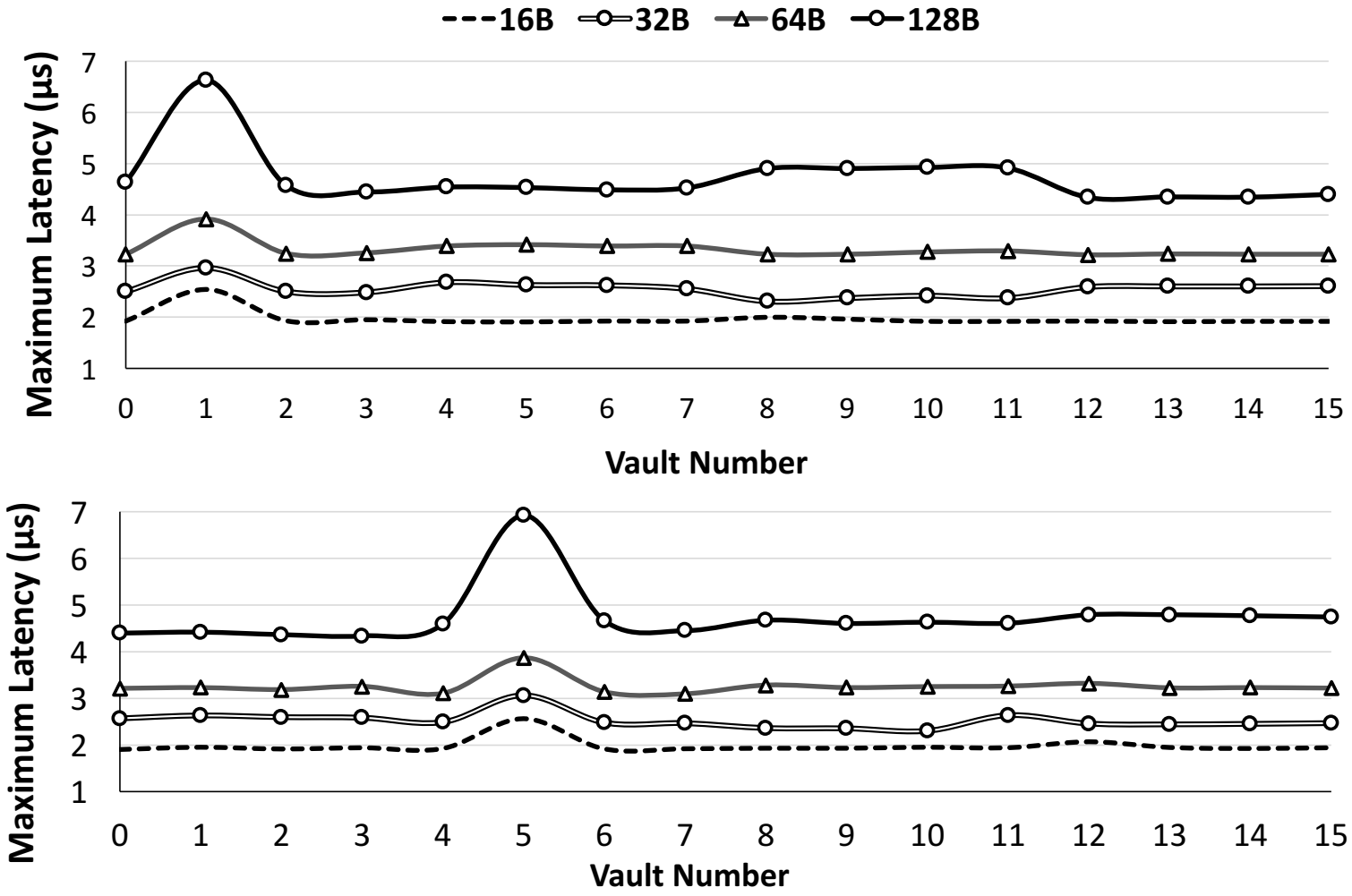
[2] Average Latency vs Requests



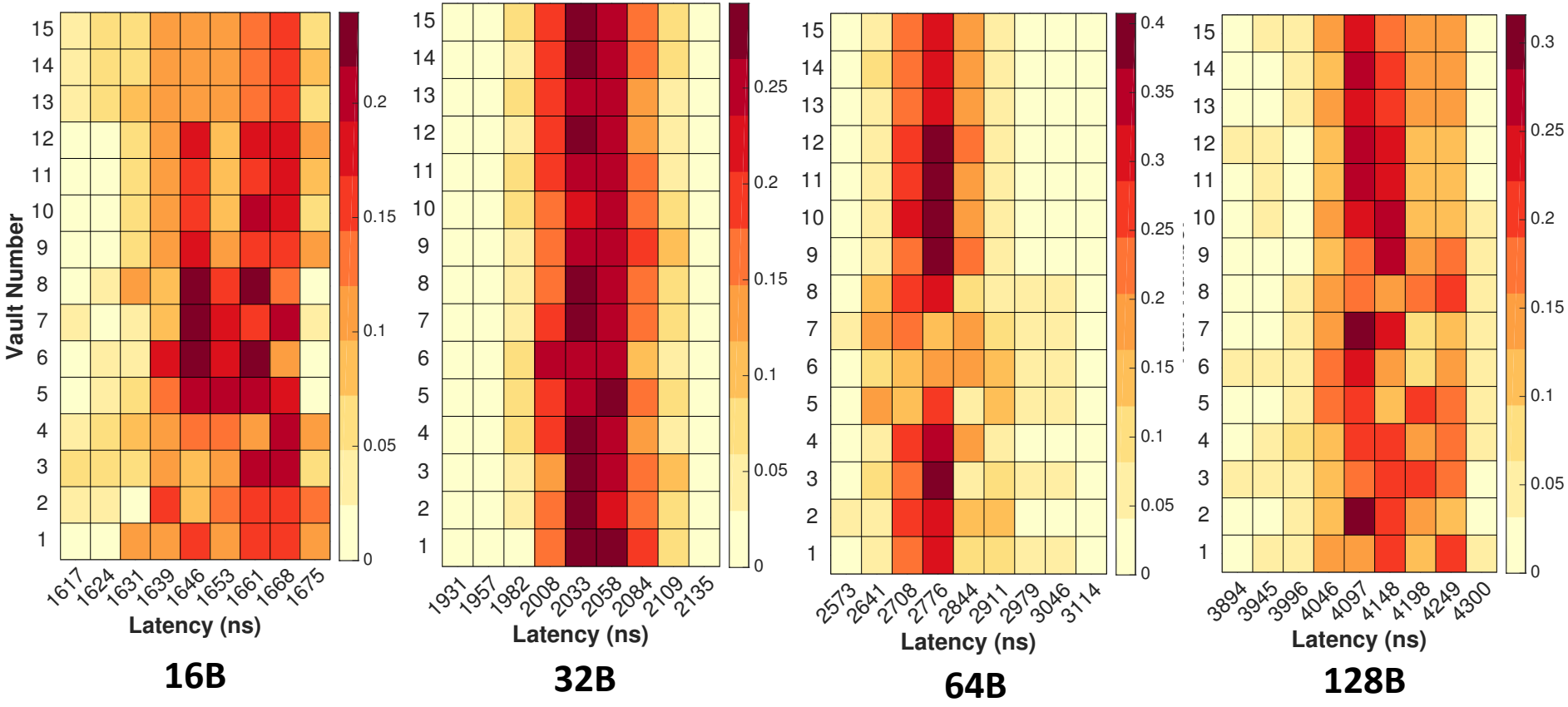
[2] Average Latency vs Requests II



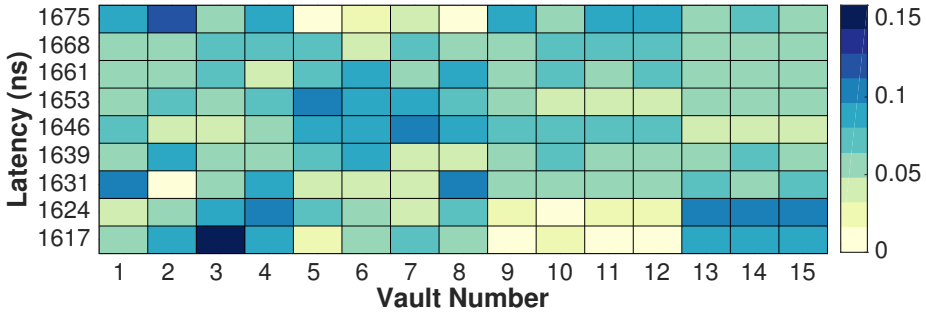
[3] QoS for 4 Vaults



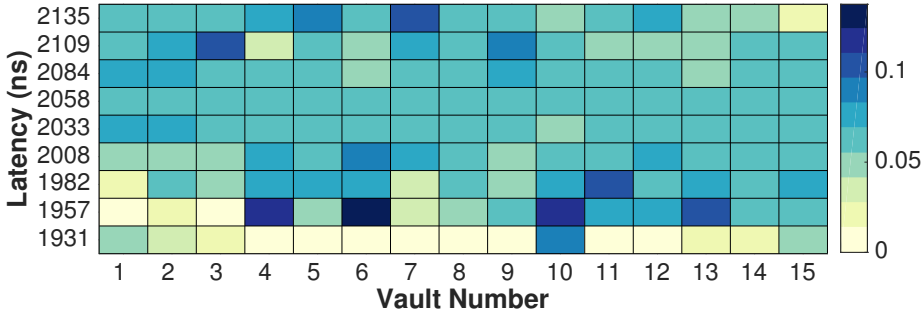
[4] Latency vs. Request Size



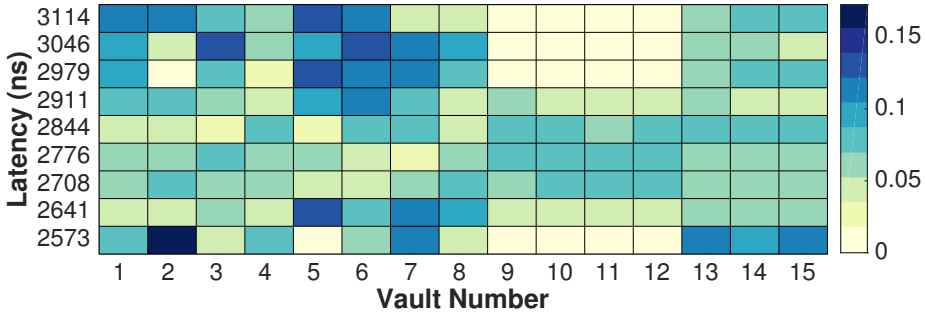
[4] Latency vs. Request Size



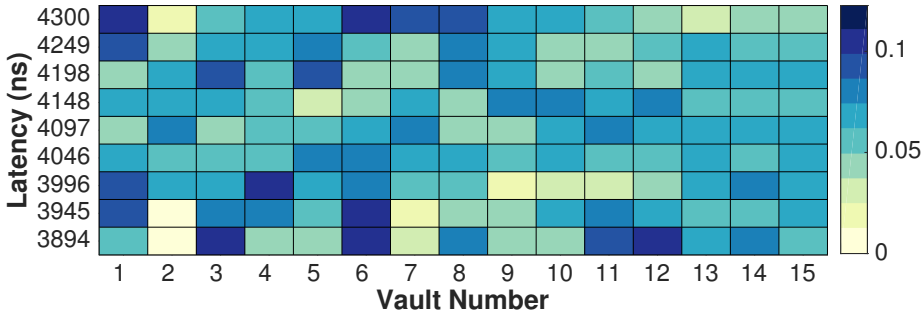
16B



32B

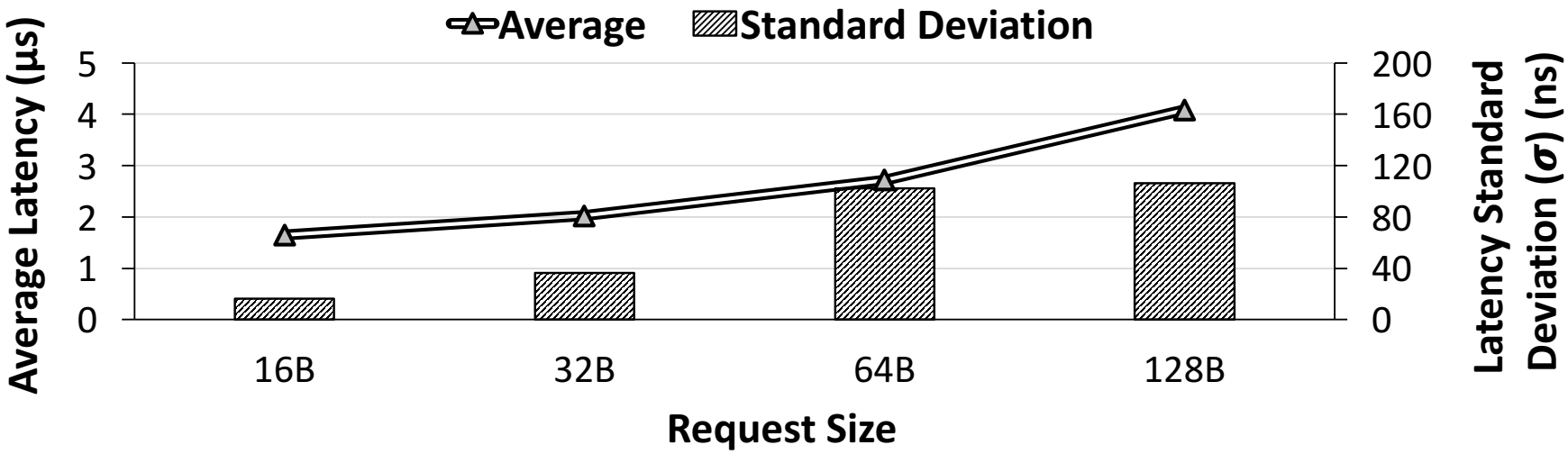


64B

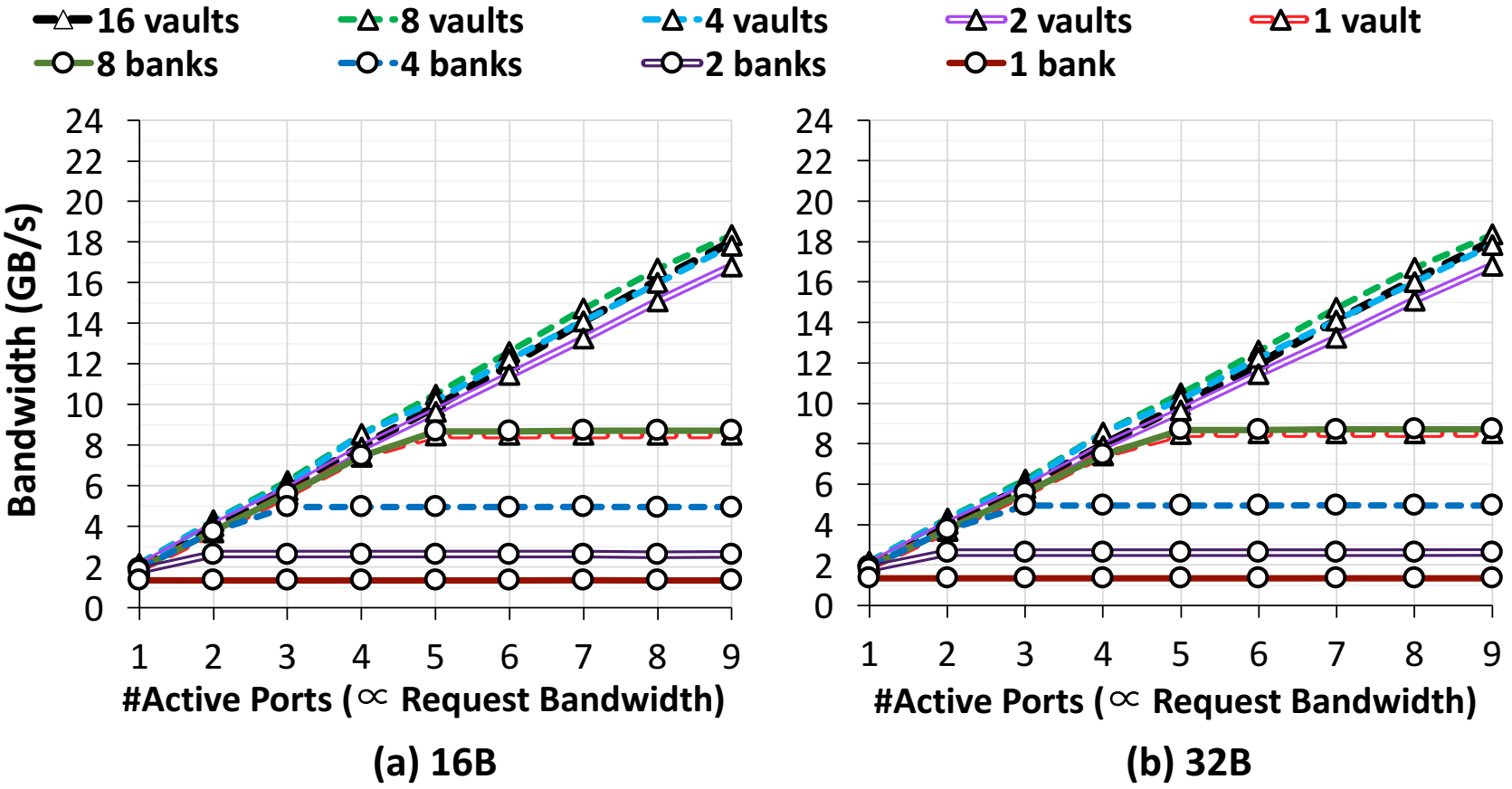


128B

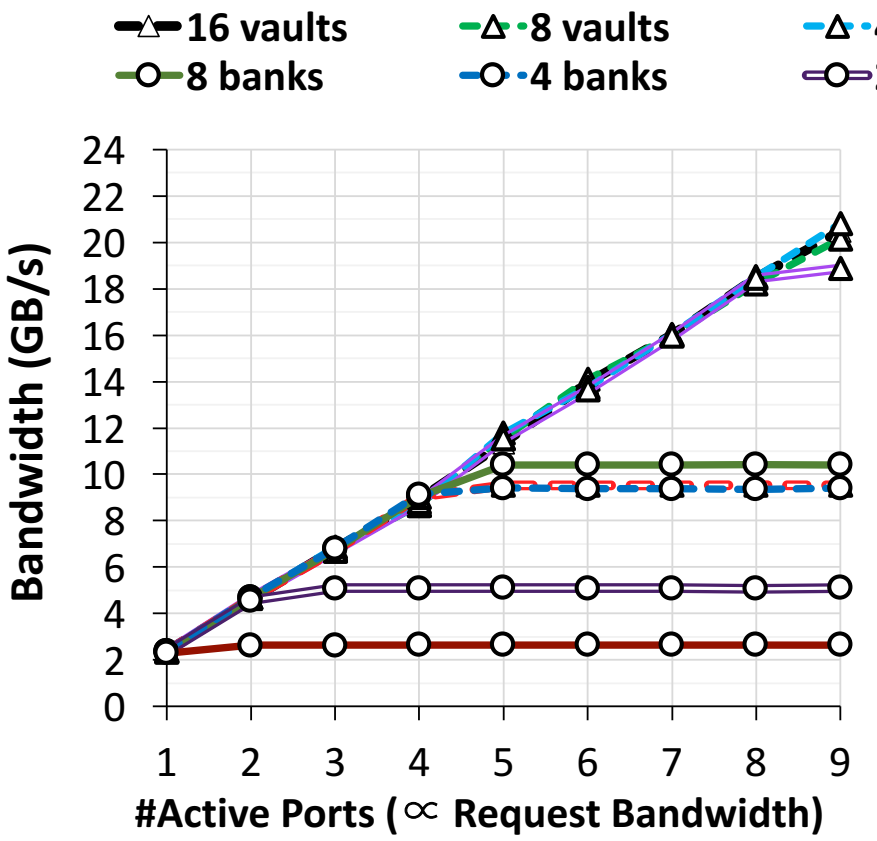
[4] Average Latency – 4 Vaults



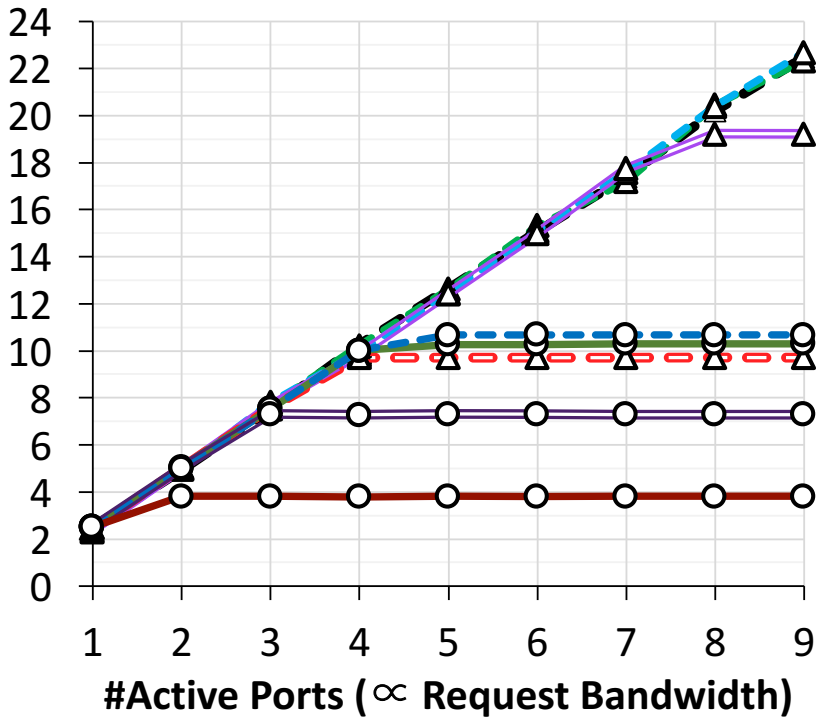
[5] GUPS – Bandwidth vs. Active Ports



[5] GUPS – Bandwidth vs. Active Ports II

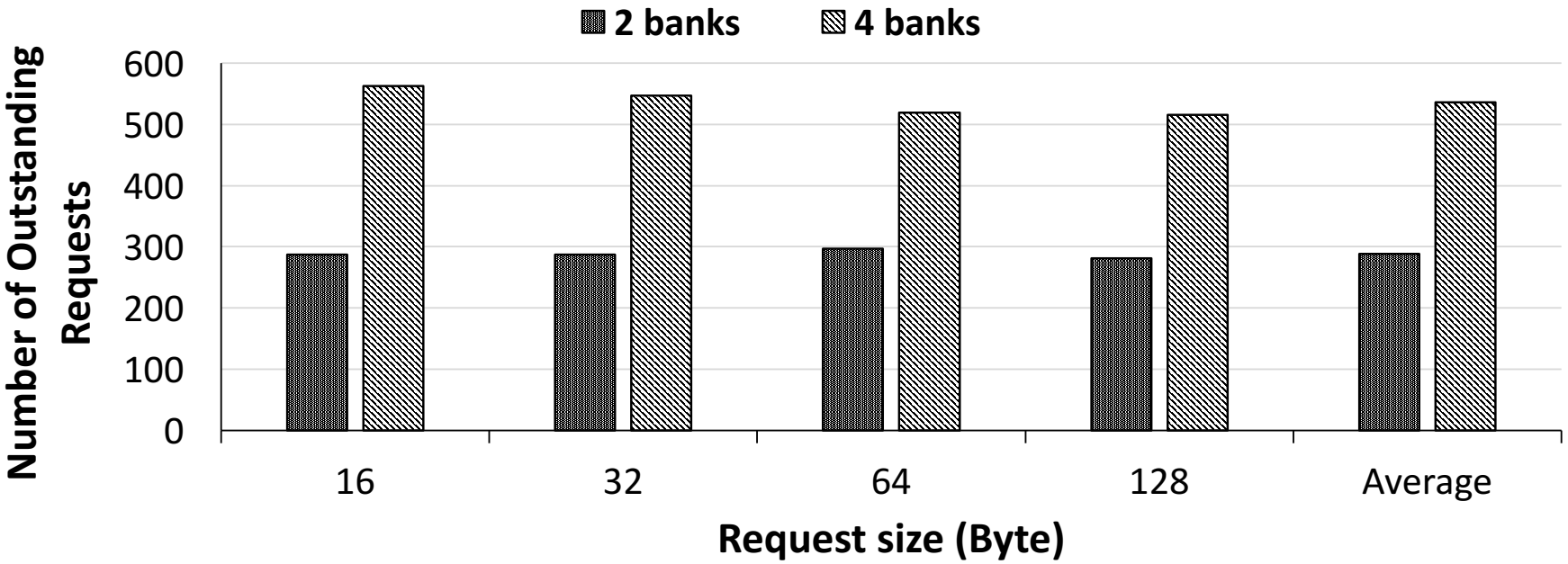


(c) 64B



(d) 128B

[6] GUPS – Outstanding Requests



Takeaways

Large and small requests allow tuning for bandwidth- or latency-optimized applications better than DRAM

- Vault- and bank-level parallelism are key to achieving higher BW

Vault latencies are more correlated with access patterns and traffic than with physical vault location

Queuing delays will continue to be a concern with NOCs in the HMC

- Address via host-side queuing/scheduling or by distributing accesses across vaults (data structures or compiler passes)

The HMC's NoC complicates QoS due to variability

- However, trade-offs in packet size and "private" vaults can improve QoS

Questions?

Thanks to Micron for helping to support our HMC testbed!