SNIA ESF | ETHERNET STORAGE

Architectural Principles for Networked Solid State Storage Access – Part 2

Today's Presenters

Doug Voigt
Chair, NVM Programming Model,
SNIA Technical Council
Distinguished Technologist, HPE

J Metz
SNIA Board of Directors
R&D Engineer
Cisco

SNIA at a glance

SNIA ESF | ETHERNET STORAGE

160
unique member companies

3,500
active contributing members

50,000
IT end users & storage pros worldwide

Learn more: **snia.org/technical**

@SNIA

SNIA Legal Notice

SNIA ESF | ETHERNET STORAGE

- The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Focus on Principles

- Technical and market dynamics are creating complexity
  - Permutations of technologies and positioning
  - Intricacies of new technology integration
- Foundational principles have not changed
  - Application views of memory and storage
  - The role of data access time (latency) in system architecture
- Principles transcend details
  - This presentation uses principles to guide detailed analysis
  - This presentation does not report benchmark results
- On-demand webcast: Architectural Principles for Networked Solid State Storage Access – Part 1 https://www.brighttalk.com/webcast/663/203821

Times are changing

- Storage access times are shrinking
  - Emerging persistent memory (PM) technologies
  - Faster than flash
- Interconnects are getting faster
  - Bandwidth
  - Latency
- Creating challenges for software
  - Software stacks are starting to dominate latency
  - Trigger for a fundamental architecture shift

Principles we will cover today

- Application views of Persistent Memory technology
- Why latency determines application view
- Latency budget analysis
- Latency and system scale

Definitions

- IO – protocol used to access storage
- Poll – repeated reading of IO state to detect completion
- Context Switch – allow other processes to use a core
- Load/Store (Ld/St) – CPU instructions that access memory
- Non-Uniform Memory Access (NUMA) – describes a memory system that exhibits a significant range of latencies due to underlying technology or scale

SNIA ESF | ETHERNET STORAGE

Application View

Recap of Part 1: IO vs Ld/St

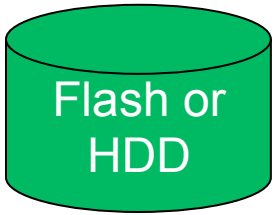
- IO
  - Data is read or written using RAM buffers
  - Software has control over how to wait (context switch or poll)
  - Status is explicitly checked by software
- Ld/St
  - Data is loaded into or stored from processor registers
  - Software is forced by processor to wait for data during instruction
  - No status checking – errors generate exceptions

Application vs technology views

Application Sees

Technology Is

IO



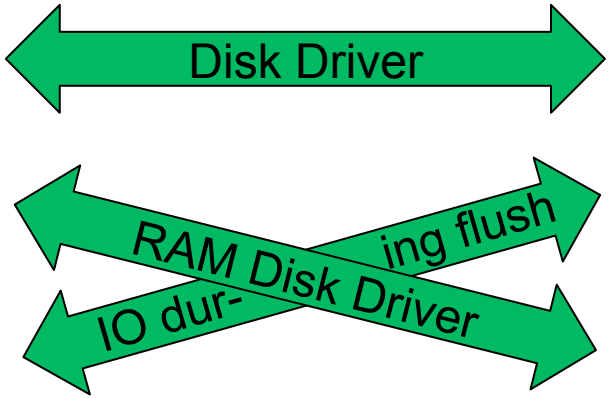
Ld/St

Application vs technology views

Application Sees

Technology Is

IO



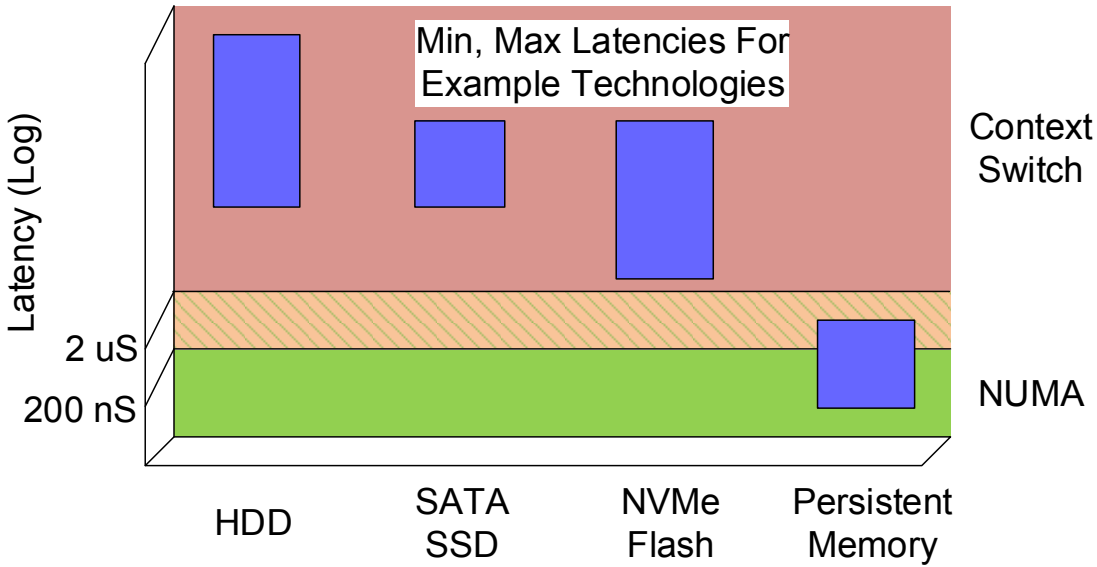
Flash or HDD

Ld/St

Processor Instructions

Persistent Memory

Latency Thresholds Cause Disruption



IO

Ld/St

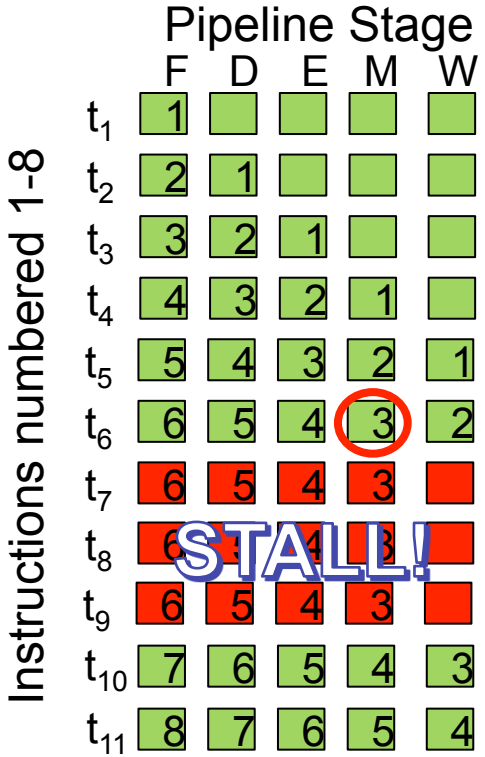
Persistent Memory – Purist Definition

- Does not require (long term) power to retain its contents
- Can be accessed using Ld/St/Mov instructions
- Without too much loss of processor throughput

When is it OK to force the CPU memory access pipeline to wait for a storage or memory access to complete?

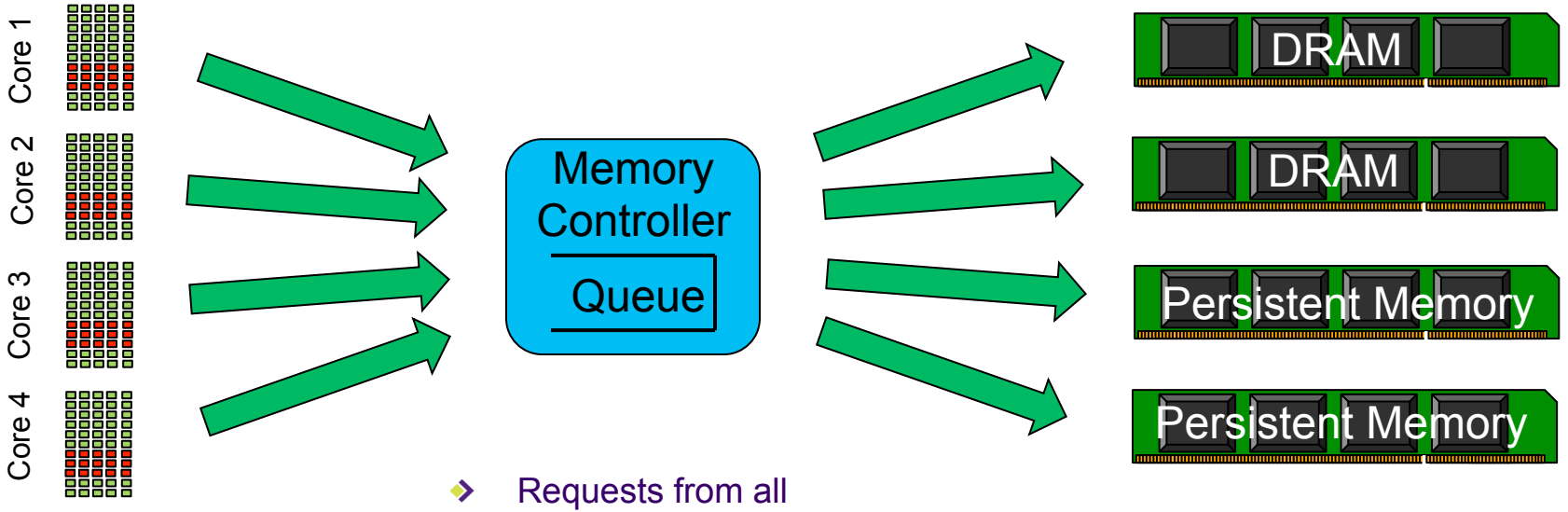
- May pause a thread in the middle of executing an instruction
- May block reads and writes to all core

Pipeline Stall Wastes Processor Throughput



- Each core's instruction pipeline has a limited number of stages (5 shown here)
- If a memory access (column M, time t_6) takes longer than the pipeline is designed for, it stalls (t_7-t_9) so processing power is wasted.
- If this happens a lot the processor can grind to a halt

Memory Pipeline Interference



- High memory latency can have domino effect across cores
- Requests from all cores intermingle
- PM requests can delay DRAM requests
- DIMMs have pipelines too
- Increase concurrency but may also increase latency

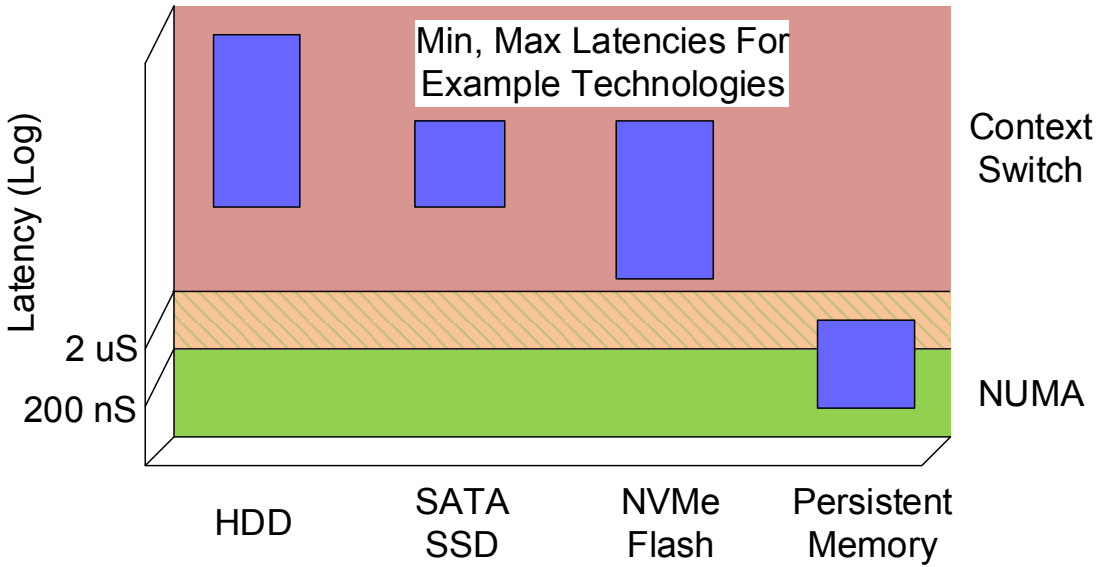
NUMA Systems are designed to mitigate these effects for bounded memory latencies

Why latency determines application view

	Pro	Con	Acceptable Latency
Ld/St	Lowest overhead	Stalls pipelines if slow	NUMA
Poll	Moderate overhead	Consumes one thread	< ~2 uS
Context Switch	High overhead	Free while blocked	> ~2 uS

- The acceptable upper bound of NUMA latency depends on processor architecture and application instruction mix
- The acceptable upper bound for polling depends on processor specific context switch time

Latency Thresholds Cause Disruption



IO

Ld/St

SNIA ESF | ETHERNET STORAGE

Latency Budgets

Latency budget building blocks

- Interconnect hops
- Media
- Host
- Queueing - throughput

Serial Interconnect Hop Latency

SNIA ESF | ETHERNET STORAGE

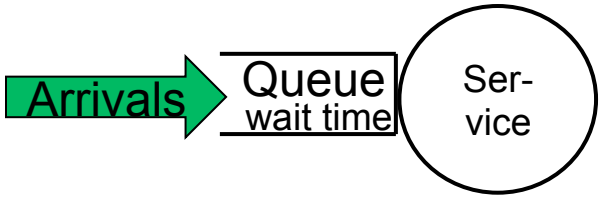
- Considerations
  - Speed of Light: .3 m/nS. (e.g. 2m = 7 nS) + 10-100nS for SERDES pair
  - Data Transfer: Xmit bit rate * (Headers + Payload) * Encoding Derating
  - Port: SERDES + 0-100 nS
  - Switching, Routing: 0-∞
- "Typical" switch latency examples
  - PCIe: >= 20 nS
  - IB: >= 90 nS
  - Ethernet: >= 300 nS
- 1 Interconnect hop = 2*Ports(>= 10 nS per pair) + <#Switches>*Switch(>=20 nS) + distance/.3m/ns(or more)

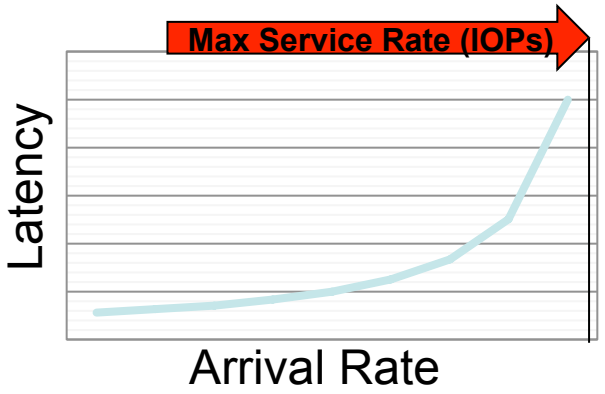
Media Latency

- Considerations
  - Command/Response HW: 10-1000 nS
  - Driver Software, Interrupt Response: 0-20 uS
  - Translation/Virtualization: 0-100 uS
  - Seek/Select/Enable: 10 nS (DRAM), 10+ uS (Flash), 1+ mS (HDD)
  - Data Transfer: Media bit rate * (Headers + Payload) * Encoding
- "Typical" Examples (single threaded)
  - 20 nS DRAM
  - 70 uS Flash
  - 1 mS HDD

Hosts and Queues

- Host Considerations
  - PCIe: Treat as additional hops for modular RNIC/HBAs
  - Driver Software: 1 – 50 uS
- Queue Considerations
  - Latency = 1/(Service Rate – Arrival Rate)

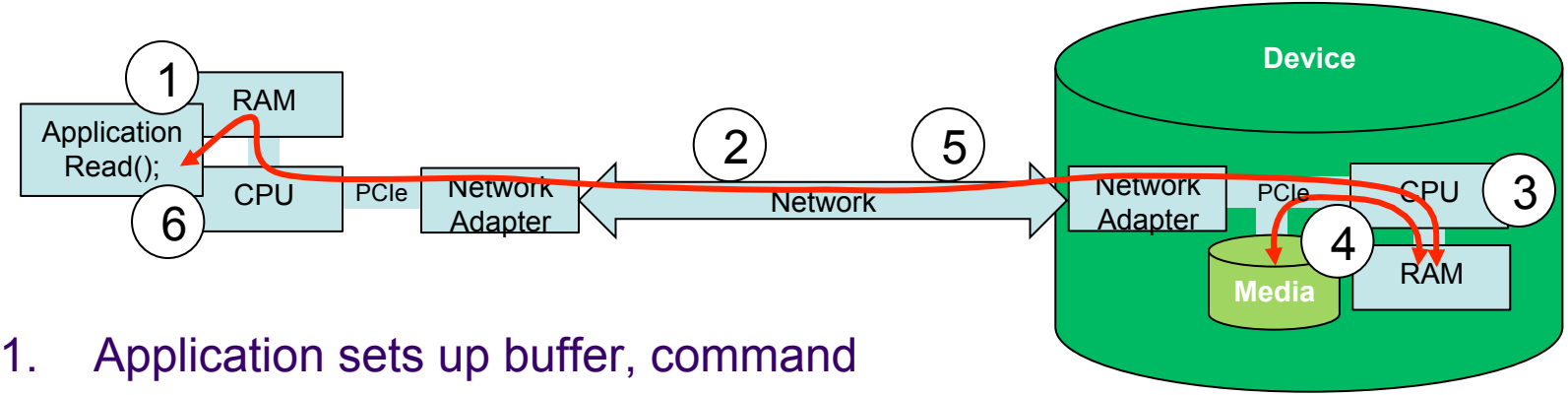




Latency budget templates

- Examples provided
  - IO
  - RDMA and Remote Persistence
  - Scale out memory
- Numerical disclaimer:
  - Template latency examples are from the building block slides above.
  - The main purpose is to enable engineers to determine where their networked PM implementations fall in the latency disruption chart.
  - Constant innovation and tuning in components and systems continues to drive latency down.

Latency template for IO



1. Application sets up buffer, command
2. Application sends command to SSD RAM
3. SSD SW processes IO
4. SSD accesses media, RAM data buffer (read)
5. SSD Sends data, response
6. Application receives response

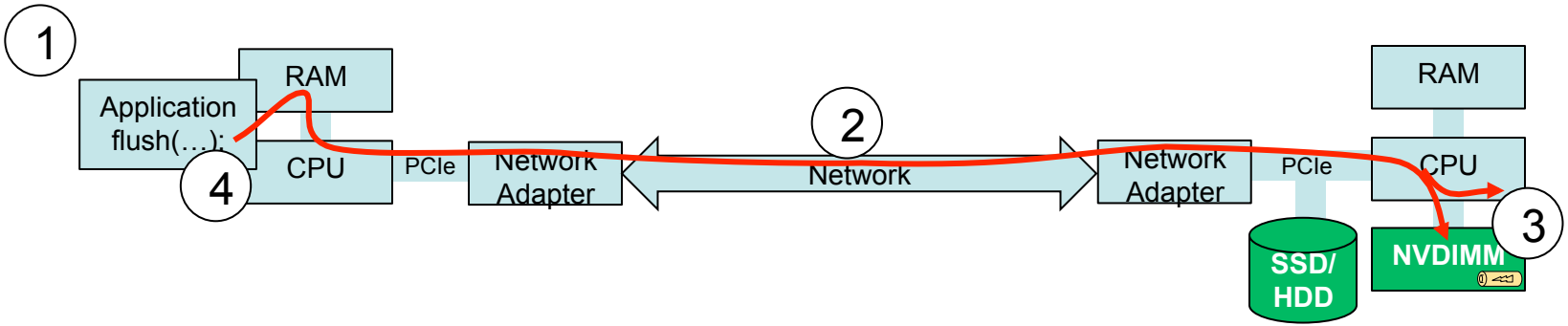
IO latency budget in uS*

	Host	Network	Device (SSD)	
1	50	1.2 - 1.6		
2		.1-.3		
3			100	
4		1.1 - 1.3		
5			see above	
6	see above			
Totals	50	1.2 - 1.6	100	152

- 1K data at 1 Gby
- 1 Switch
- Moderate load
- All units in uS

*Rough Approximations

Latency template for RDMA write to PM



1. Application establishes RDMA connection during mMap
2. Application executes 1 or more RDMA writes during flush
3. Application executes RDMA send to force remote flush
4. Application receives response from remote flush

For more on RDMA see:
https://www.brighttalk.com/webcast/663/185909

RDMA Write to PM budget in uS*

	Host	Network	Device (PM)	Device (CPU)	
1	NA	NA	NA		
2	10	1.1 - 1.3	2		
3	20	.1 - .3	above	20	
4	above	.1 - .3		above	
Totals	30	1.3 - 1.9	2	20	54

- 1K data at 1 Gby
- 1 Switch
- Single RDMA write
- All units in uS

*Rough Approximations

Gen-Z: A New Data Access Technology

High Bandwidth Low Latency

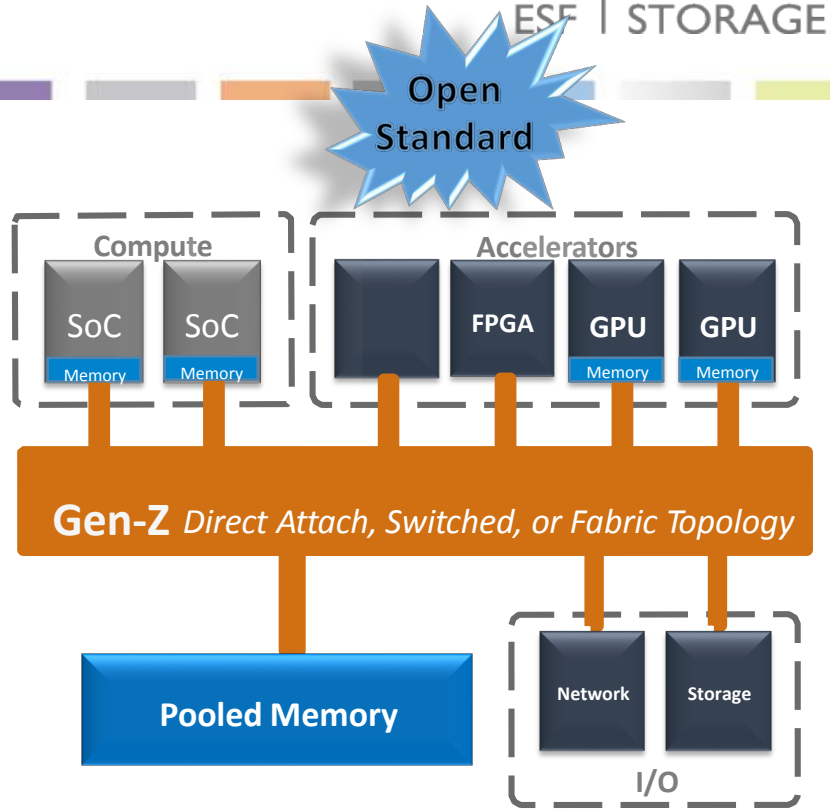
- Memory Semantics – simple Reads and Writes
- From tens to several hundred GB/s of bandwidth
- Sub-100 ns load-to-use memory latency

Advanced Workloads & Technologies

- Real time analytics
  - Enables data centric and hybrid computing
- Scalable memory pools for in memory applications
- Abstracts media interface from SoC to unlock new media innovation

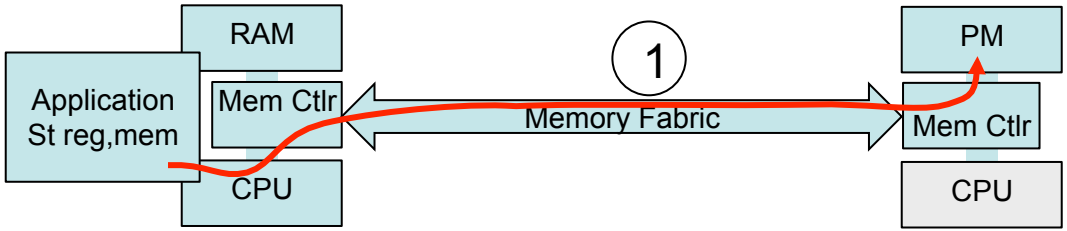
Secure Compatible Economical

- Provides end-to-end secure connectivity from node level to rack scale
- Supports unmodified OS for SW compatibility
- Graduated implementation from simple, low cost to highly capable and robust
- Leverages high-volume IEEE physical layers and broad, deep industry ecosystem



Rack Scale

Latency Template for Scale Out Memory e.g. Gen-Z



1. Application uses St instruction to write to remote memory

For more on RDMA see:
https://www.brighttalk.com/webcast/663/185909

Scale out memory budget in uS*

	Host	Network	NVDIMM	
1	.01	.2	.1	
Per St Total	.01	.2	.1	
16 Lines	.16	3.2	1.6	5

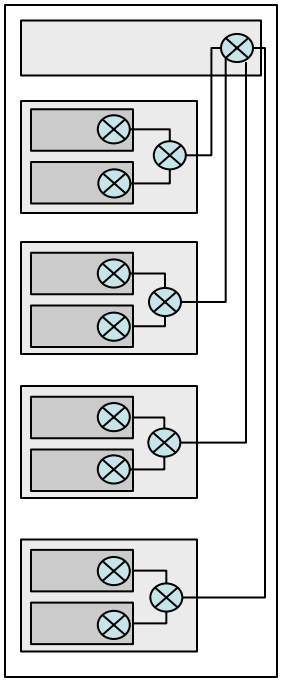
- 64 By cache line per St at 1 Gby
- 1 Switch
- Consider using mov or put/get

- 16 St's for 1K
- All units in uS

*Rough Approximations

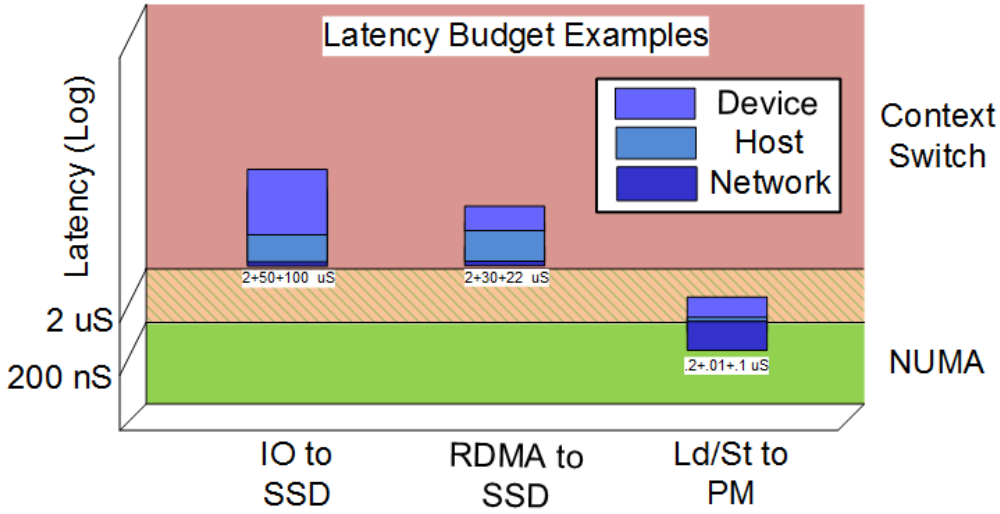
Latency contributors due to system scale

- Transmission Distance
  - Realistically communication between adjacent racks requires about 5 meters of optical cable.
  - This takes 17 nS.
- Switch/Router hops
  - Switches appear at blade, chassis and rack levels. Each contributes 100-300 nS latency totaling .5-1.5 uS



Adding it all up

- Ld/St Inhibitors
  - networks
  - > rack scale memory fabric
  - media technology mismatch
- High Ld/St Latency Pain depends on workload
  - PM access size
  - PM access mix
  - Flush



Other Helpful Resources

SNIA ESF | ETHERNET STORAGE

- On-demand Webcasts:
  - Architectural Principles for Networked Solid State Storage Access – Part 1 https://www.brighttalk.com/webcast/663/203821
  - Everything You Wanted to Know about Storage But Were too Proud to Ask: Part Teal – Buffers, Queues & Caches https://www.brighttalk.com/webcast/663/241275
  - Storage Performance Benchmarking: Solution Under Test https://www.brighttalk.com/webcast/663/164335
- SNIA NVM Programming Model: http://www.snia.org/sites/default/files/technical_work/final/NVMProgrammingModel_v1.1.pdf
- SNIA PM White Papers: https://www.snia.org/education/whitepapers

After This Webcast

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Ethernet Storage Forum (ESF) website and available on-demand
- www.snia.org/forums/esf/knowledge/webcasts
- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-ESF blog: sniaesfblog.org
- Follow us on Twitter @SNIAESF

SNIA ESF | ETHERNET STORAGE

Thank you!