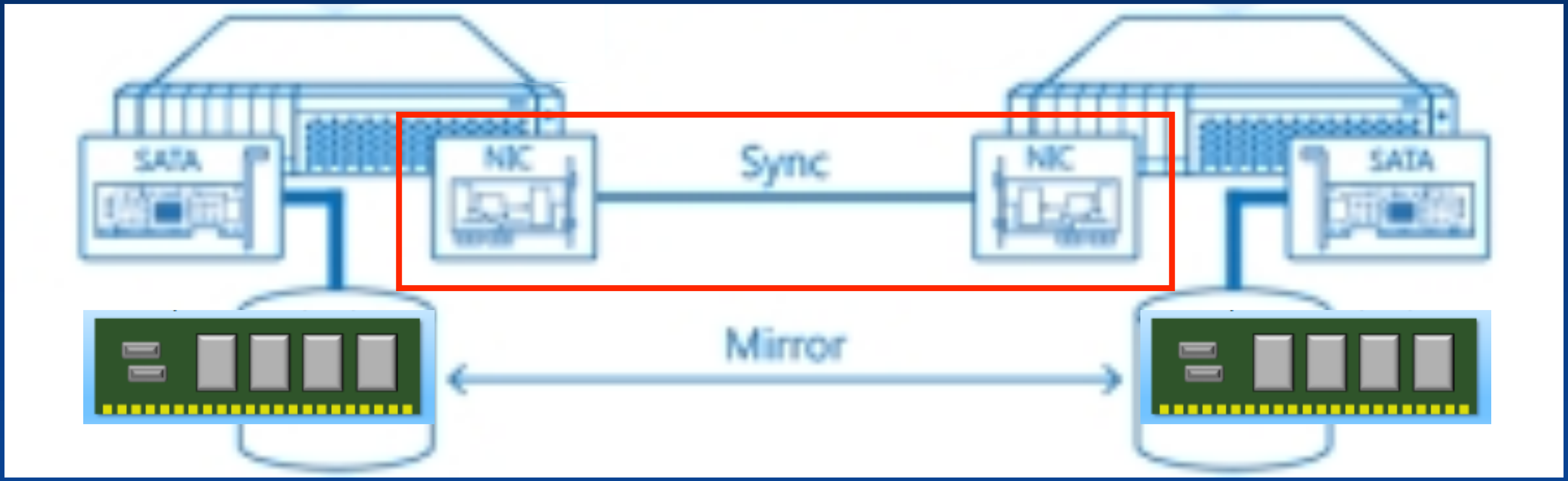
Persistent Memory(PM) over Fabrics

Rob Davis
VP Storage Technology
Mellanox

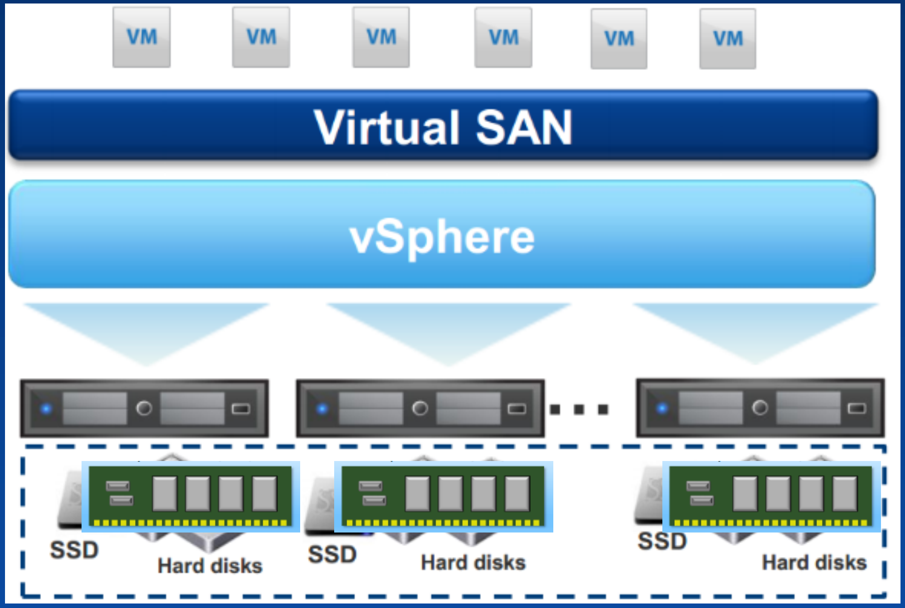
PM Needs High Performance Fabric for Storage



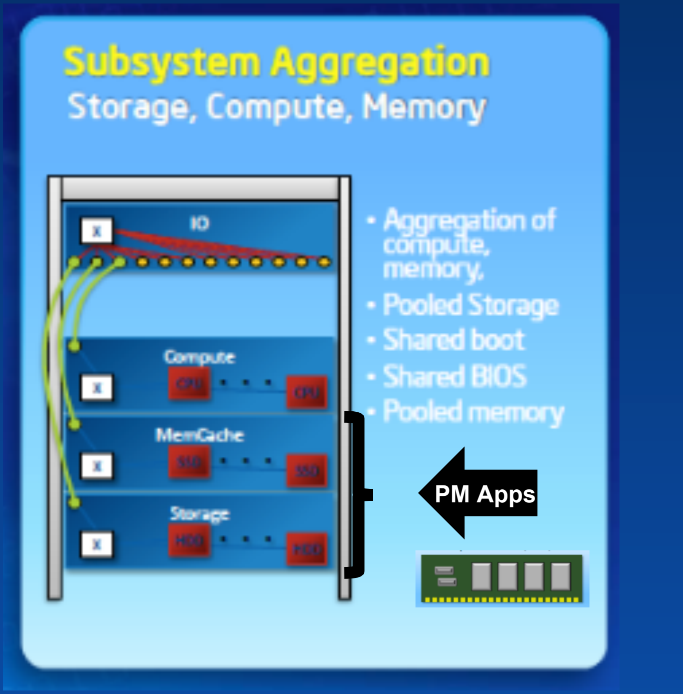
	PM	NAND
Read Latency	~100ns	~30us
Write Latency	~500ns	~500us

Networked PM Applications

Hyper Converged



Disaggregation


Subsystem Aggregation

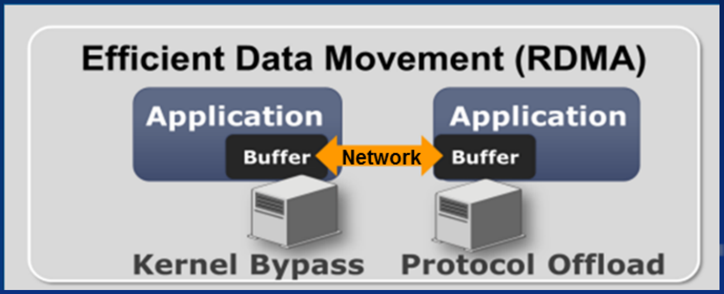
Storage, Compute, Memory

- Aggregation of compute, memory,
- Pooled Storage
- Shared boot
- Shared BIOS
- Pooled memory

How Do We Get There

- Standards Work in Progress
  - SNIA NVM Programming Model TWIG
  - SNIA NVM PM Remote Access for High Availability
  - Other standards efforts
- Protocol:
  - RDMA is the obvious choice
  - 2015 FMS demos:
    - PMC(7us 4KB IO)
    - HGST(2.3us)

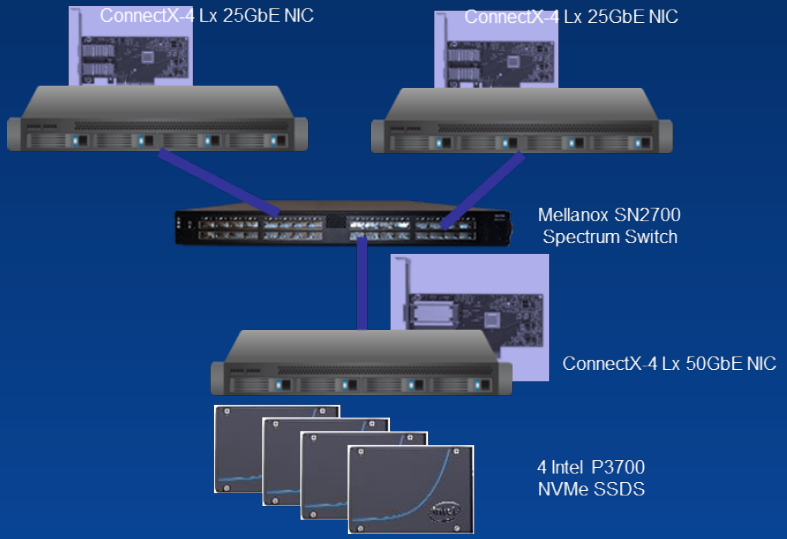
What is RDMA?


Efficient Data Movement (RDMA)
Application
Buffer
Network
Application
Buffer
Kernel Bypass
Protocol Offload

NVMf Recent (Q2 2016) Performance With Community Drivers

Topology –

- Two compute nodes
  - ConnectX4-LX 25Gbps port
- One storage node
  - ConnectX4-LX 50Gbps port
  - 4 X Intel NVMe devices (P3700/750 series)
- Nodes connected through switch

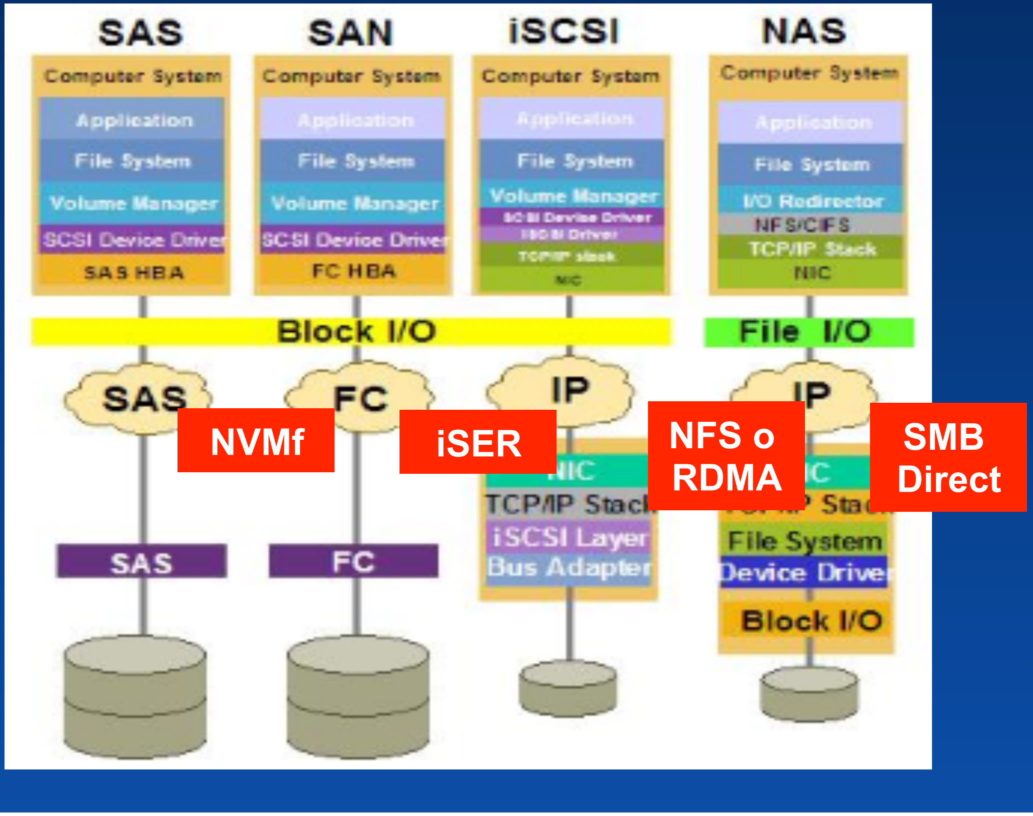


Added fabric latency
~12us

	Bandwidth (Target side)	IOPS (Target side)	Num. Online cores	Each core utilization
BS = 4KB, 16 jobs, IO depth = 64	5.2GB/sec	1.3M	4	50%

Remote PM Extensions Must be Implemented in Many Places

Data needs to be pushed to Target



NVMf

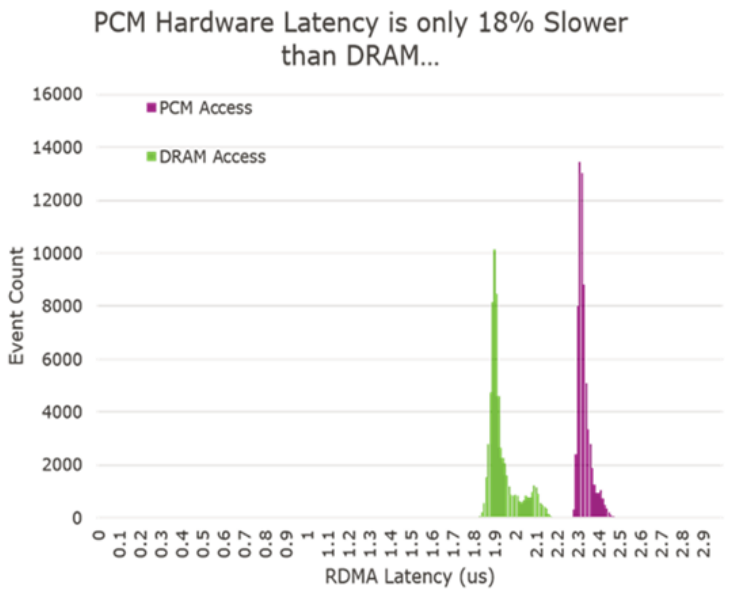
iSER

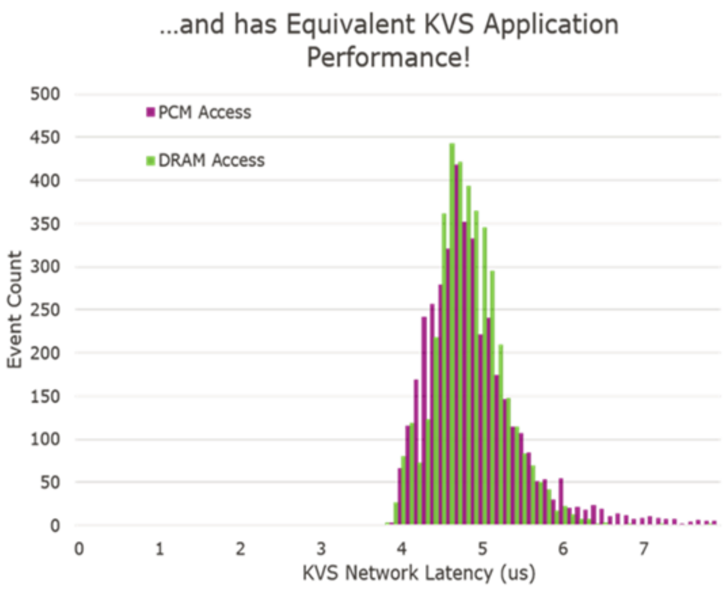
NFS o RDMA

SMB Direct

Remote Cache Flush Needed

Equivalent performance with App.


PCM Hardware Latency is only 18% Slower than DRAM...


...and has Equivalent KVS Application Performance!

https://www.hgst.com/company/media-room/press-releases/HGST-to-Demo-InMemory-Flash-Fabric-and-Lead-Discussions

Summary

- PM is great technology but needs to be networkable to achieve its full potential
- RDMA seems the obvious protocol choice
- Remote cache flush and the ability to push data to the targets is needed
- Exact parity with local DRAM performance may not be required

Thanks!

robd@Mellanox.com