Innovative NVMe MR-IOV Solution

Brian Pan

CEO, **H3 Platform Inc.**

Agenda

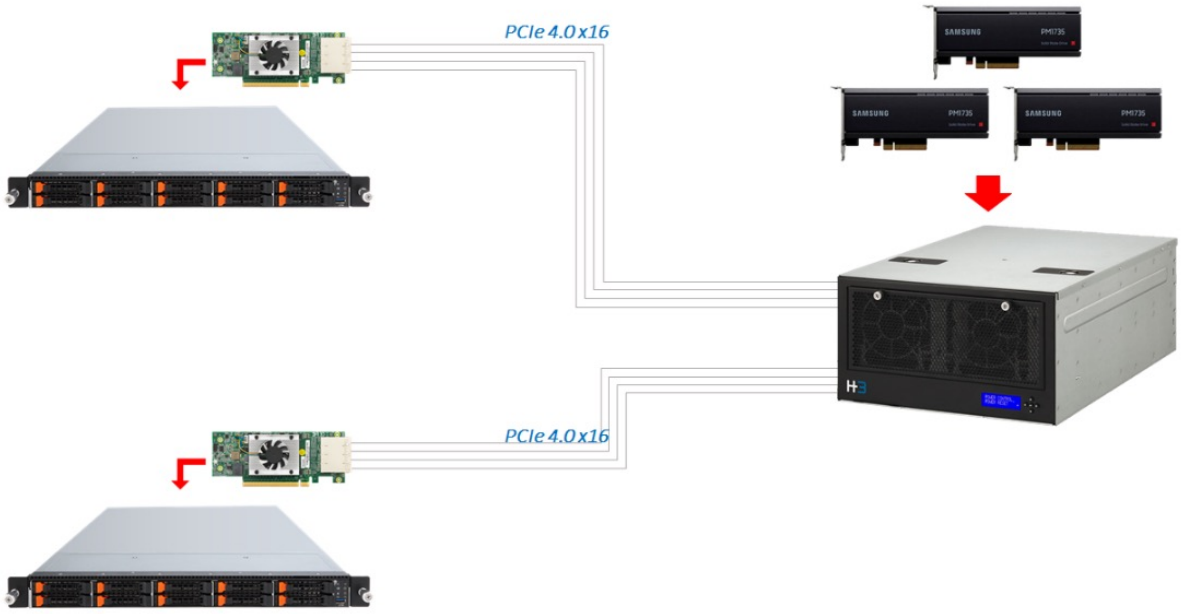
- System Architecture
- Sharing NVMe SSD in Fabrics
- Performance Testing Results
- Key Benefits
- Implementation Challenges

System Architecture

Falcon 5208
NVMe MR-IOV Solution

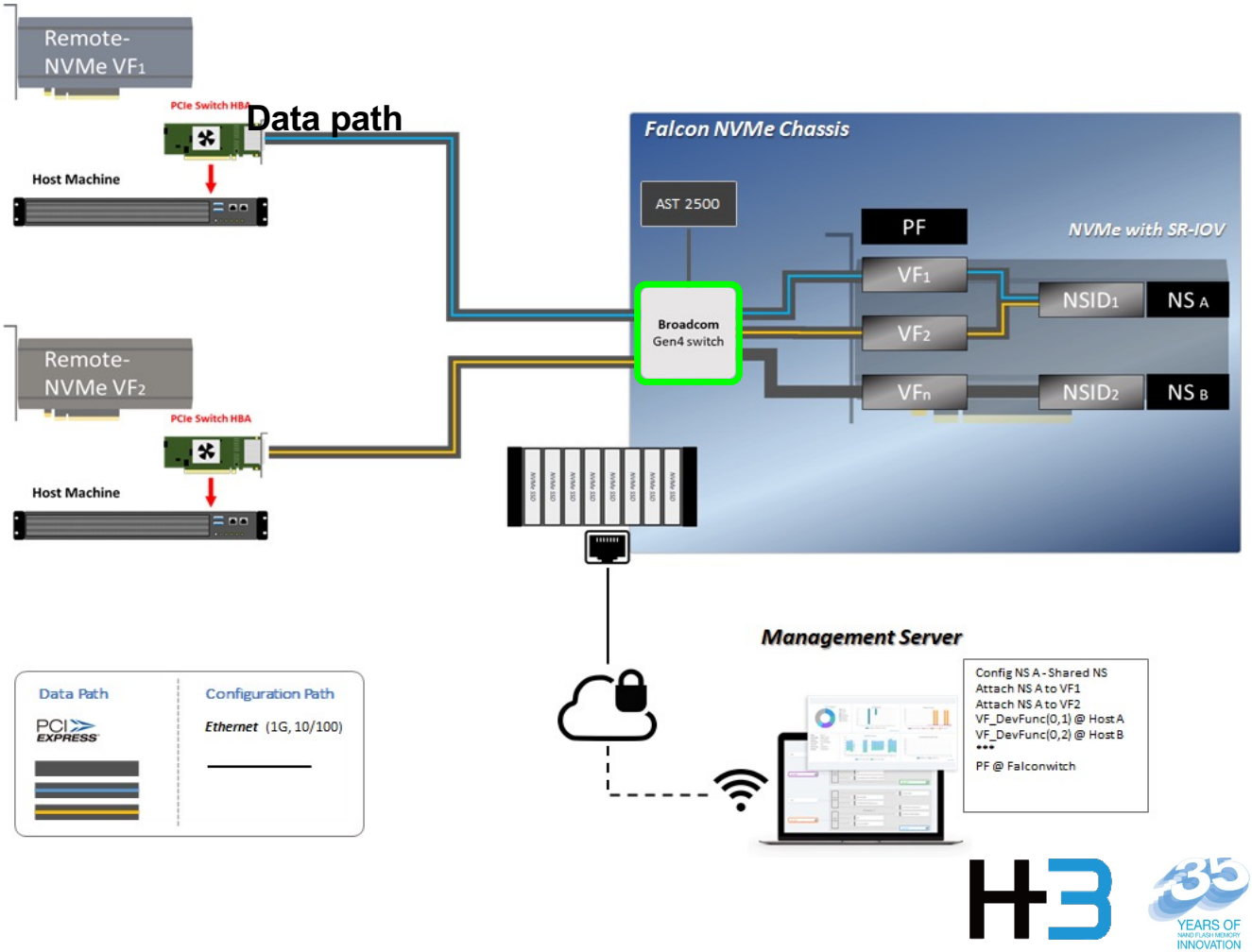
Falcon 5208

- Multi-host connectivity.
- Enables standard SR-IOV
- Assignable NVMe controllers. (Virtual functions)
- Sharable NVMe Namespace



Solution Architecture

- PCIe Fabric
- NVMe end-to-end
- SSD config. Management



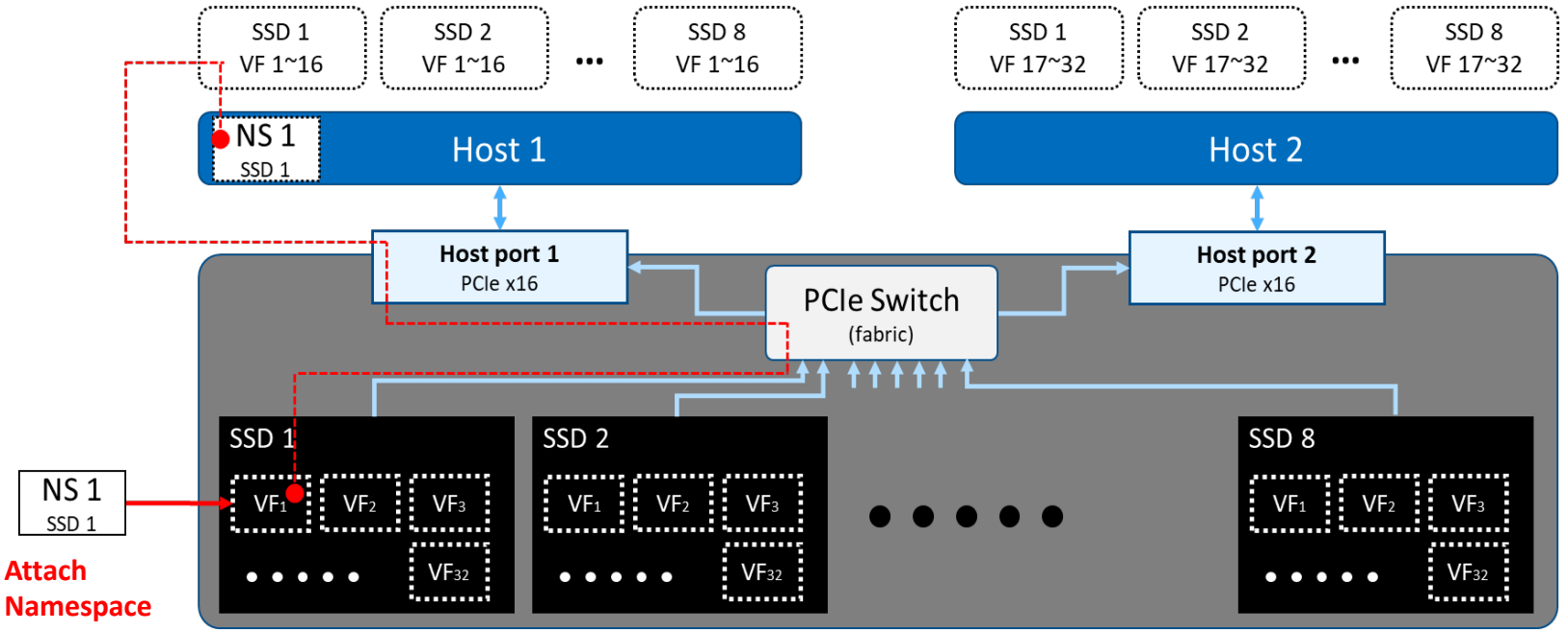
Sharing NVMe SSD in Fabrics

Virtual Functions, Shared Namespaces

Assign Multi-Logic NVMe SSDs

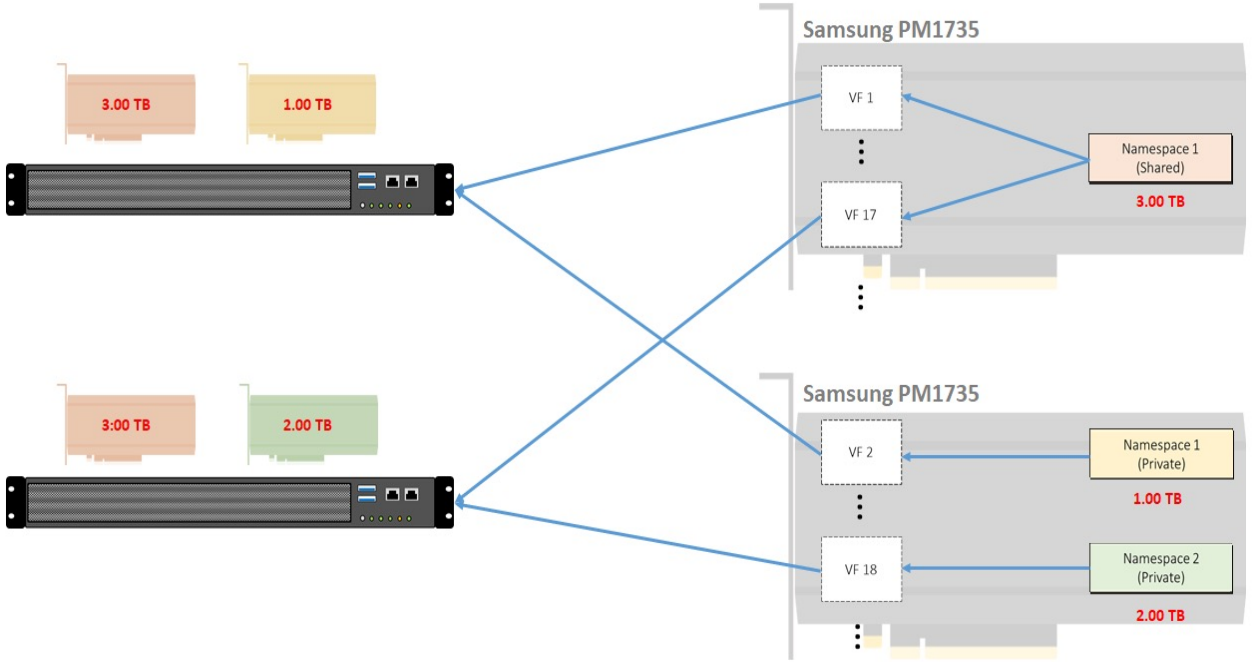
The **Virtual Function** (NVMe Controller) are assigned to connected hosts.

The **Namespaces** (NVM resources) are attached to the VFs.



Shared Namespaces

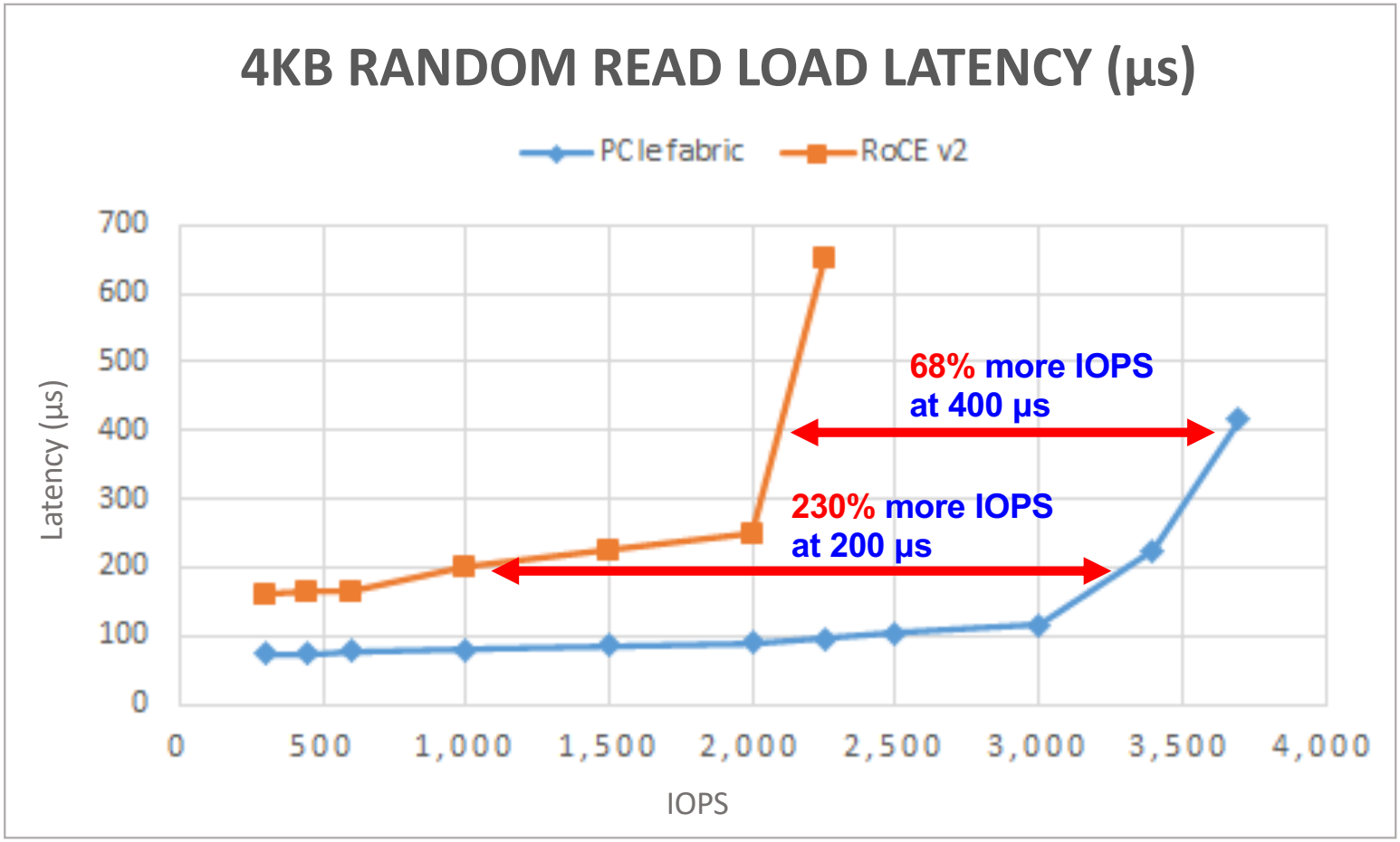
- The shared namespace can be attached to multiple VFs.
- There are 256 namespaces and VFs in Falcon 5208.
- Standard SR-IOV bypass Hypervisor to eliminate overhead.



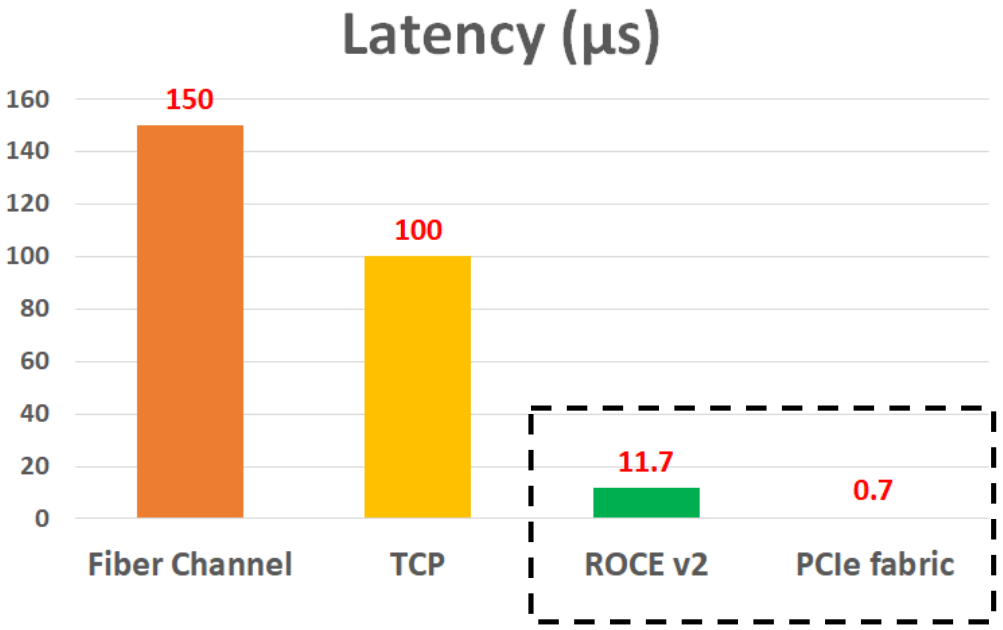
Performance Testing Results

IOPS, Latency

IOPS – PCIe Fabric vs RoCE v2



Latency of Fabrics



PCIe has lowest latency and highest throughput

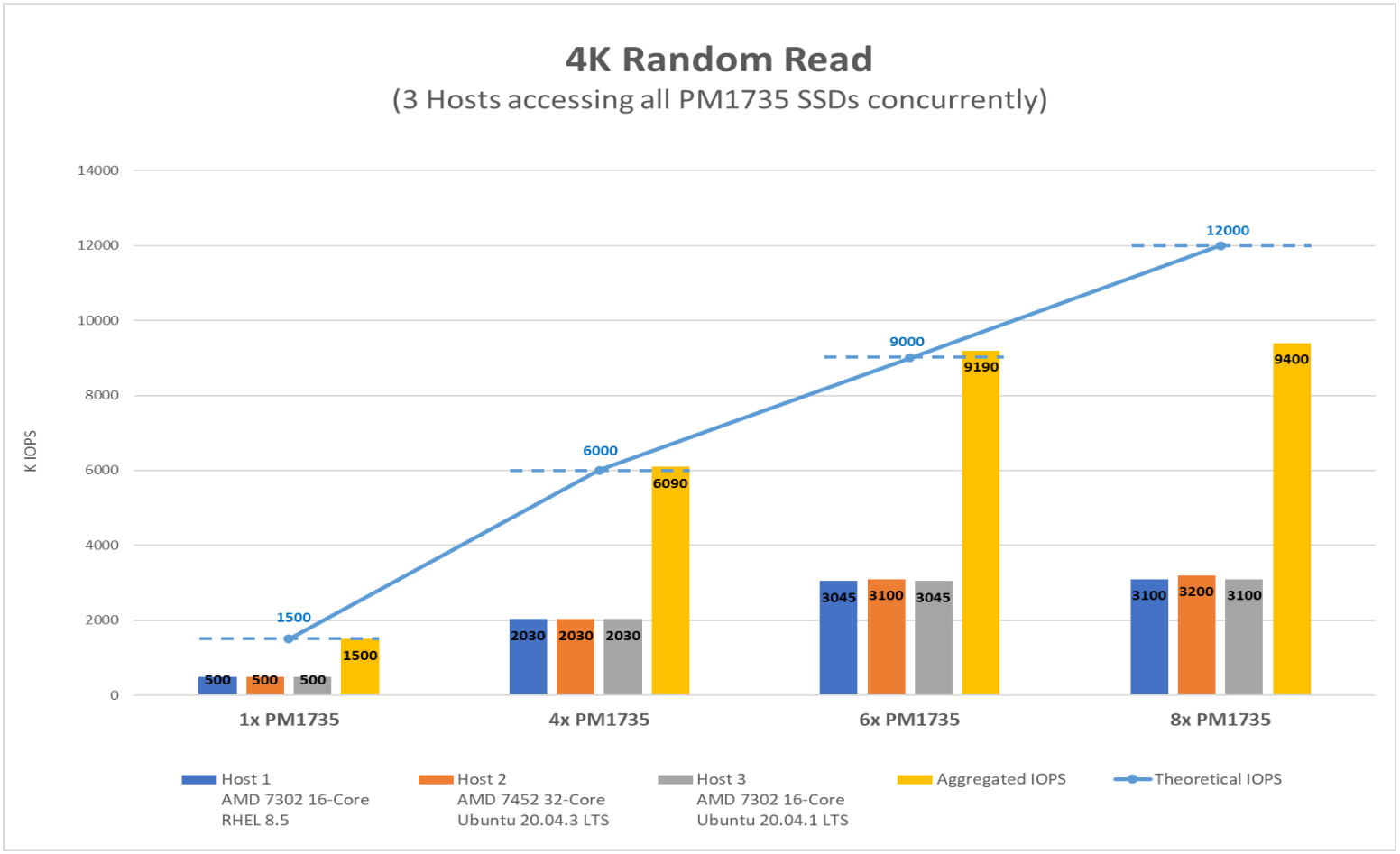
- 0.7 µs fabric latency
- 48 GB/s throughput (Gen 4)
- 12M IOPS of the VFs sharing among the hosts

4K Random Read: Multi-host

- 3 Hosts are accessing all PM1735 SSDs concurrently.
- Scales to 6x NVMe SSDs

	Host_1 (x8) AMD 7302 16-Core RHEL 8.5	**Host_2 (x16)** AMD 7452 32-Core Ubuntu 20.04.3 LTS	**Host_3 (x8)** AMD 7302 16-Core Ubuntu 20.04.1 LTS	**Remark**
1x 1735	500K	500K	500K	Spec: 1,500K Result: 1,500K
4x 1735	2,030K	2,030K	2,030K	Spec: 6,000K Result: 6,090K
6x 1735	3,045K	3,100K	3,045K	Spec: 9,000K Result: 9190K
8x 1735	3,100K	3,200K	3,100K	Spec: 12,000K Result: 9,400K

4K Random Read: Multi-host (Graph)

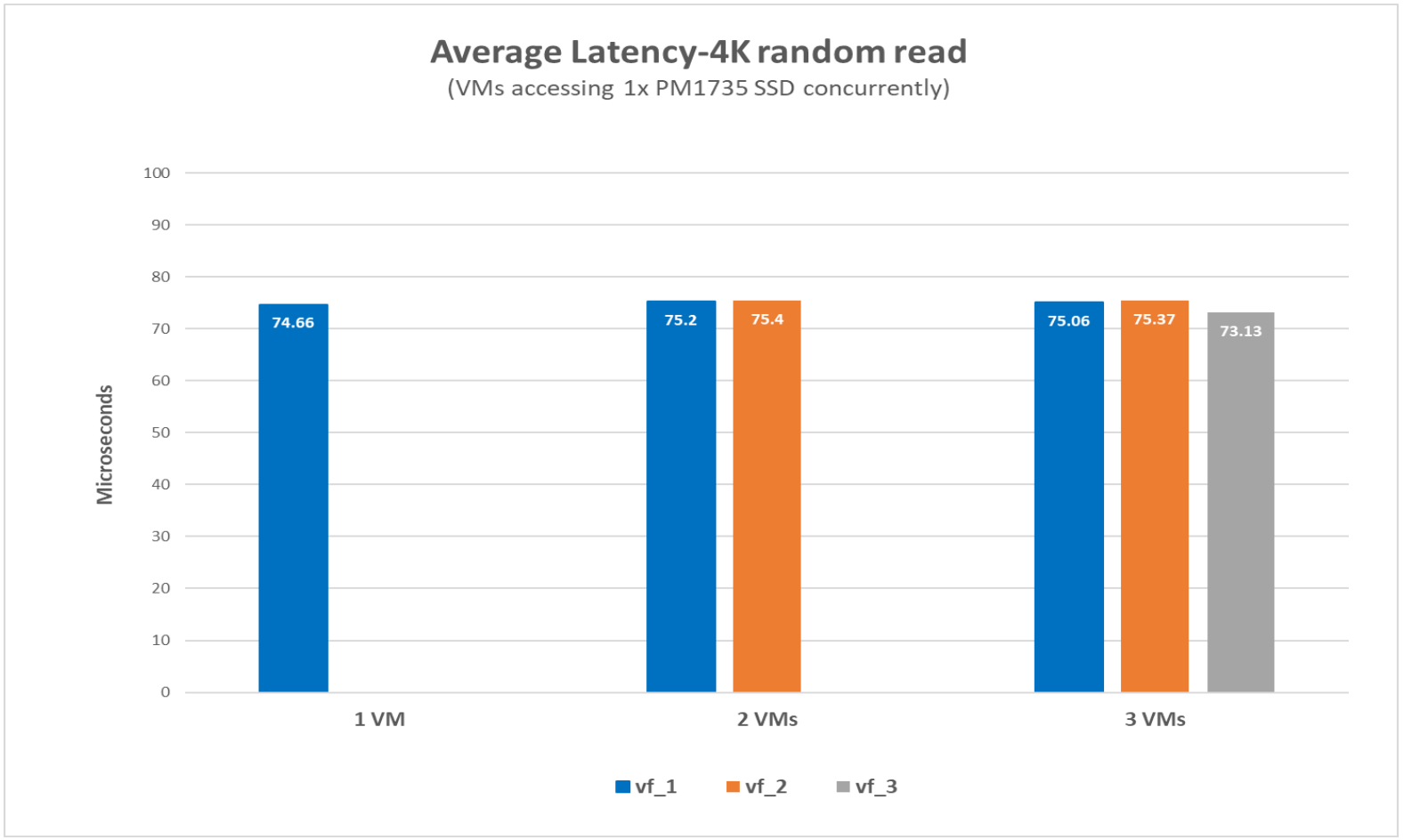


VM Access Latency-Random Read

- Latency - 4K random read
- Same as bare-metal host access.

	VM_1 (8 CPUs)	VM_2 (8 CPUs)	VM_3 (8 CPUs)	
1x 1735 (1 VF)	avg=74.66 lat (usec) : 50=0.01%, 100=99.99%, 250=0.01%	--	--	all private NS
1x 1735 (2 VFs)	avg=75.20 lat (usec) : 100=99.71%, 250=0.29%	avg=75.40 lat (usec) : 50=0.01%, 100=99.70%, 250=0.30%	--	all private NS
1x 1735 (3 VFs)	avg=75.06 lat (usec) : 50=0.01%, 100=99.42%, 250=0.58%	avg=75.37 lat (usec) : 50=0.01%, 100=99.41%, 250=0.59%, 500=0.01%	avg=73.13 lat (usec) : 50=0.01%, 100=99.44%, 250=0.56%, 750=0.01%	all private NS

VM Access Latency-Random Read (Graph)

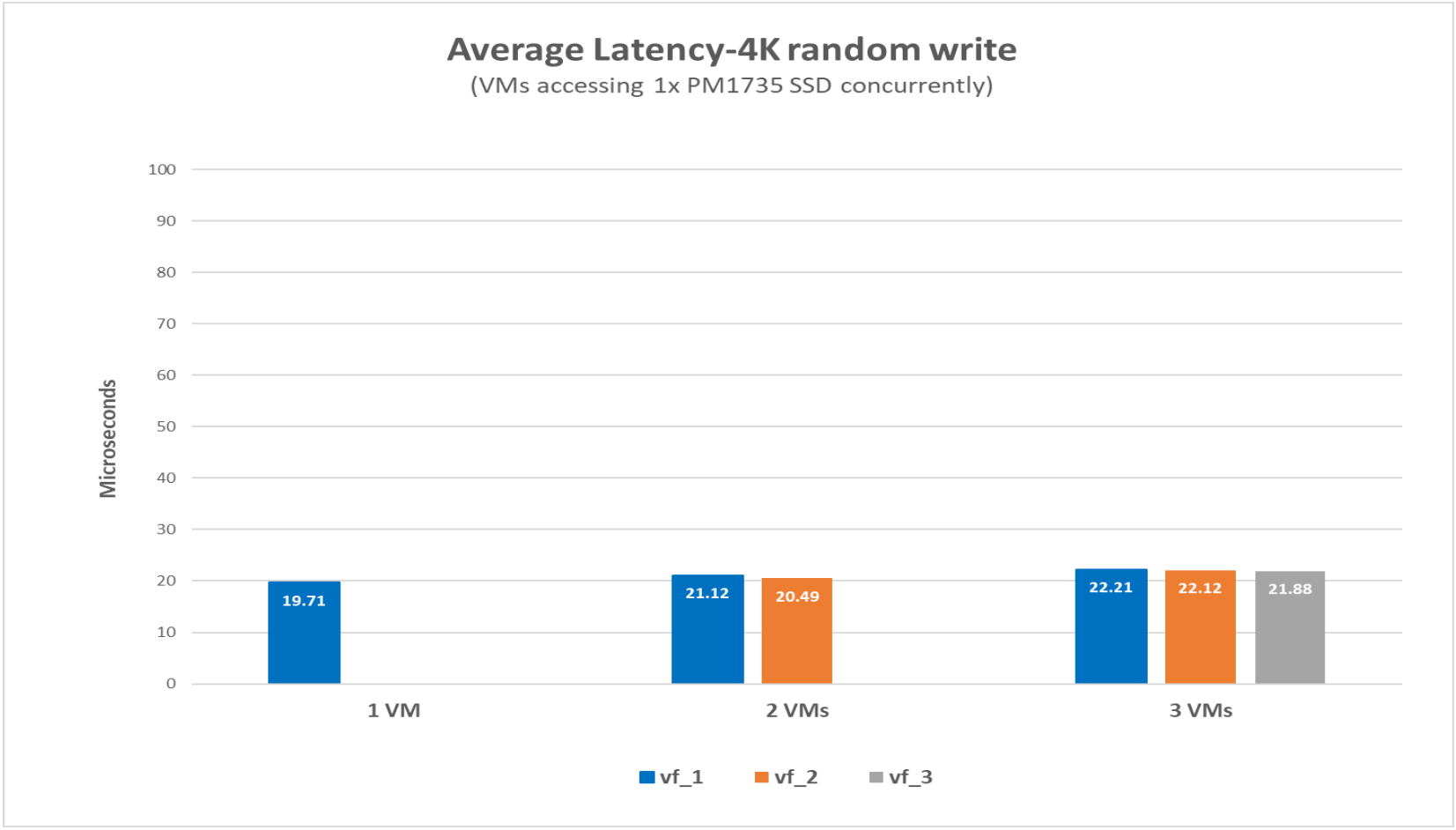


VM Access Latency-Random Write

- Latency - 4K random write
- Same as bare-metal host access.

	VM_1 (8 CPUs)	VM_2 (8 CPUs)	VM_3 (8 CPUs)	
1x 1735 (1 VF)	avg=19.71 lat (usec) : 10=0.01%, 20=96.22%, 50=3.68%, 100=0.10%, 500=0.01%			avg=432.76, **job=8** **iodepth=32**
1x 1735 (2 VFs)	avg=21.12 lat (usec) : 10=0.01%, 20=88.68%, 50=11.03%, 100=0.29%, 250=0.01%	avg=20.49 lat (usec) : 10=0.01%, 20=90.23%, 50=9.49%, 100=0.28%, 500=0.01%		avg=866.14, (vm1) avg=847.50, (vm2) **job=8** **iodepth=32**
1x 1735 (3 VFs)	avg=22.21, lat (usec) : 10=0.01%, 20=73.13%, 50=26.34%, 100=0.52%, 250=0.01%	avg=22.12, lat (usec) : 10=0.01%, 20=70.82%, 50=28.67%, 100=0.51%, 250=0.01%	avg=21.88, lat (usec) : 4=0.01%, 10=0.01%, 20=78.27%, 50=21.23%, 100=0.49%	avg=1301.73, avg=1295.08, avg=1261.72, **job=8** **iodepth=32**

VM Access Latency-Random Write (Graph)

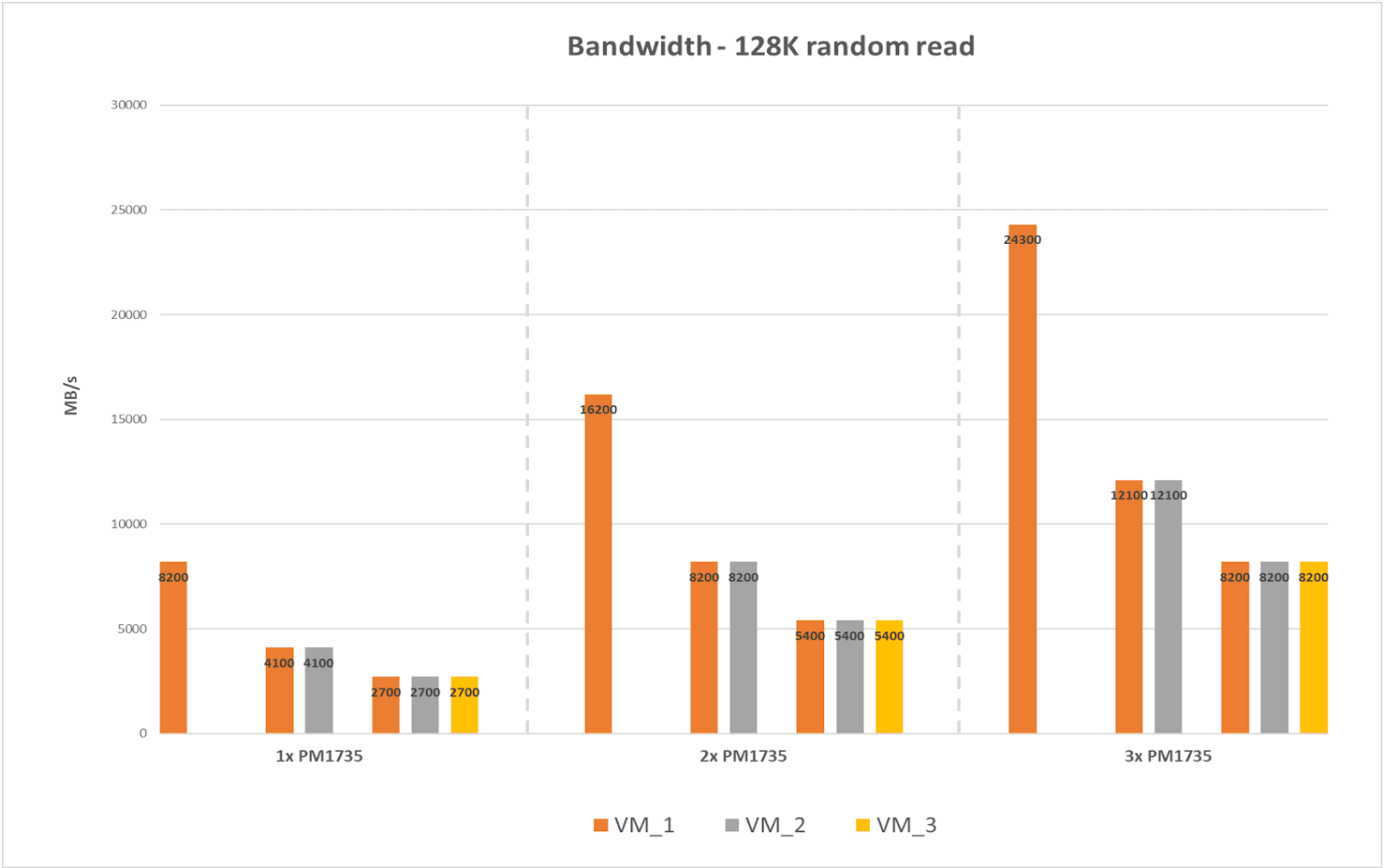


Throughput Distribution

- Bandwidth- 128K random read.
- Throughput is evenly distributed.

	VM_1 (8 CPUs)	VM_2 (8 CPUs)	VM_3 (8 CPUs)	
1x 1735 (3 VFs)	8,200 MB	--	--	all private NS
1x 1735 (3 VFs)	4,100 MB	4,100 MB	--	all private NS
1x 1735 (3 VFs)	2,700 MB	2,700 MB	2,700 MB	all private NS
2x 1735 (6 VFs)	16,200 MB	--	--	all private NS
2x 1735 (6 VFs)	8,200 MB	8,200 MB	--	all private NS
2x 1735 (6 VFs)	5,400 MB	5,400 MB	5,400 MB	all private NS
3x 1735 (9 VFs)	24,300 MB	--	--	all private NS
3x 1735 (9 VFs)	12,100 MB	12,100 MB	--	all private NS
3x 1735 (9 VFs)	8,200 MB	8,200 MB	8,200 MB	all private NS

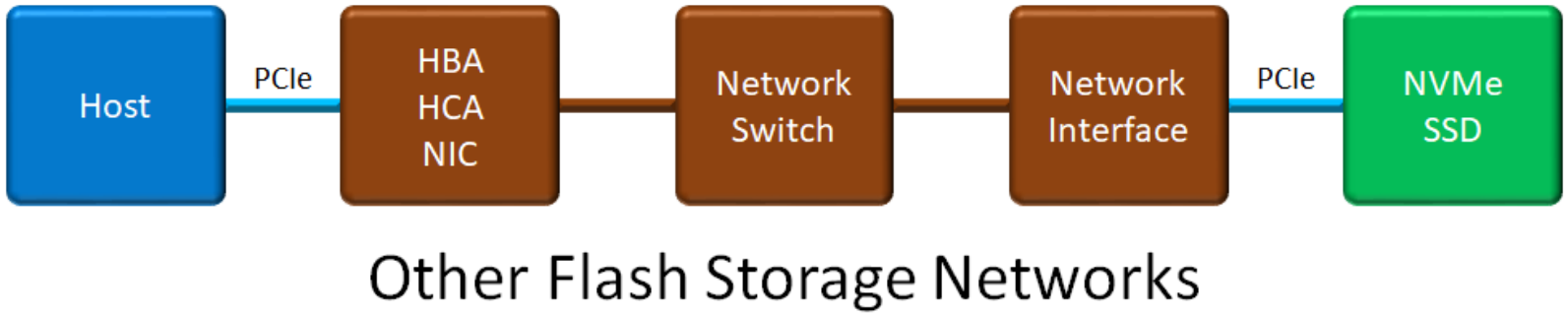
Throughput Distribution



Key Benefits

Protocol, Future Proof, Security, TCO, Flexibility

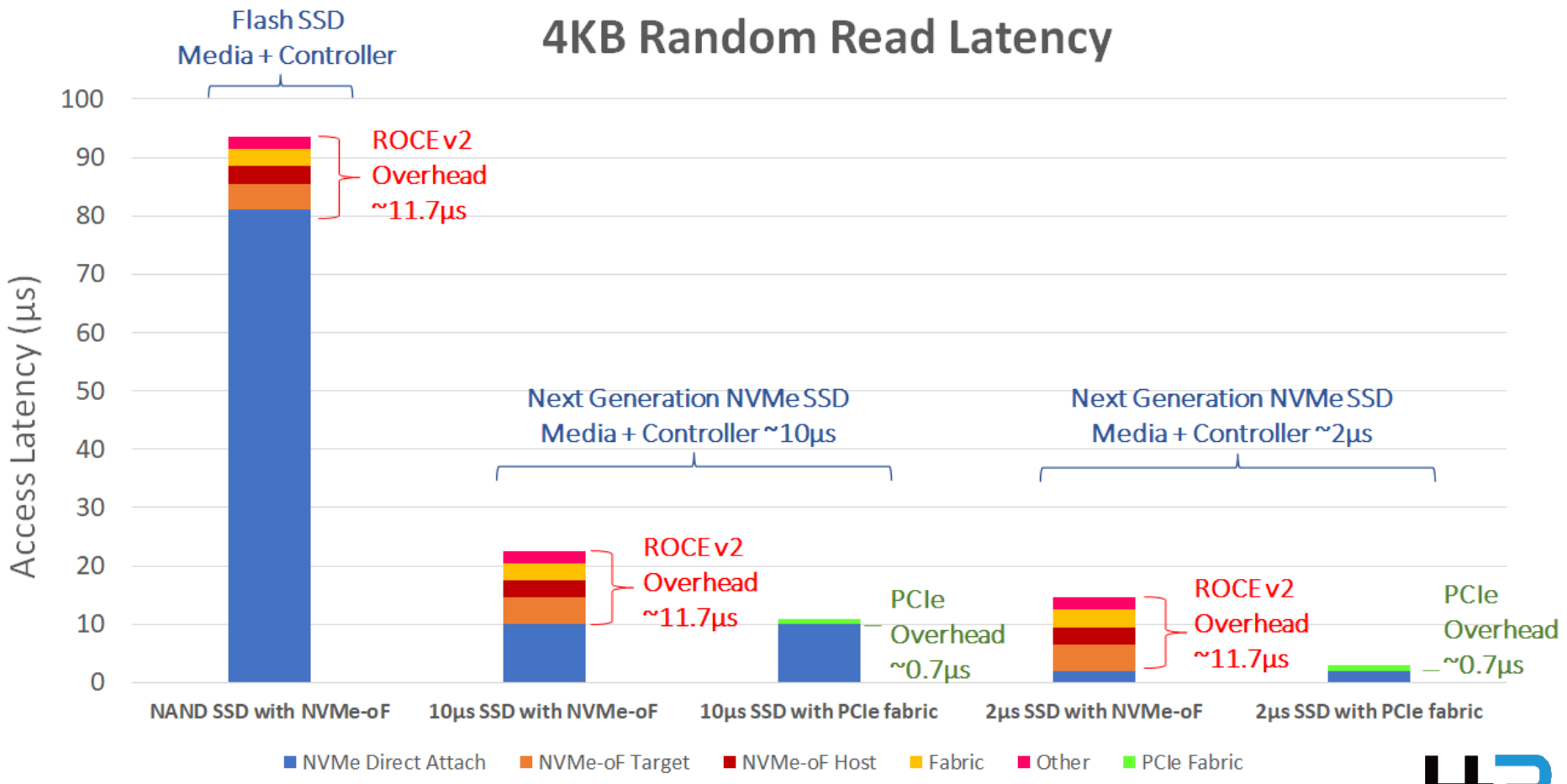
No Protocol Transfer



Other Flash Storage Networks

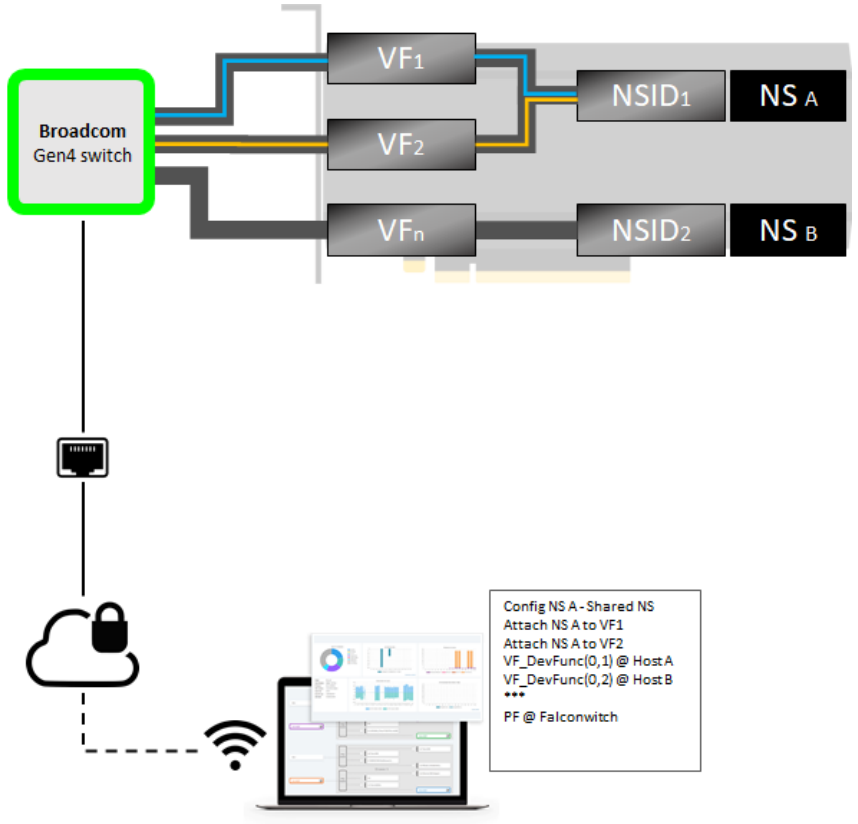
PCIe Fabric

Future Proof



Security

- **VF isolation**
  - The host can only see the assigned VFs and namespaces.
- **PF Security**
  - NVMe PF is installed on the SoC of PCIe switch, managed by the isolated management network.
  - The PF cannot read the namespace.



Lower TCO

- **Mitigate CPU Usage**
  - With SR-IOV, data is sent from storage media to virtual machines bypassing hypervisor, driving up IOPS.
- **Higher CPU / system utilization rate**
  - Minimizing the consumptions of CPU and other system resources.
  - CPU and RAM can be used for meaningful tasks rather than handling data traffics.

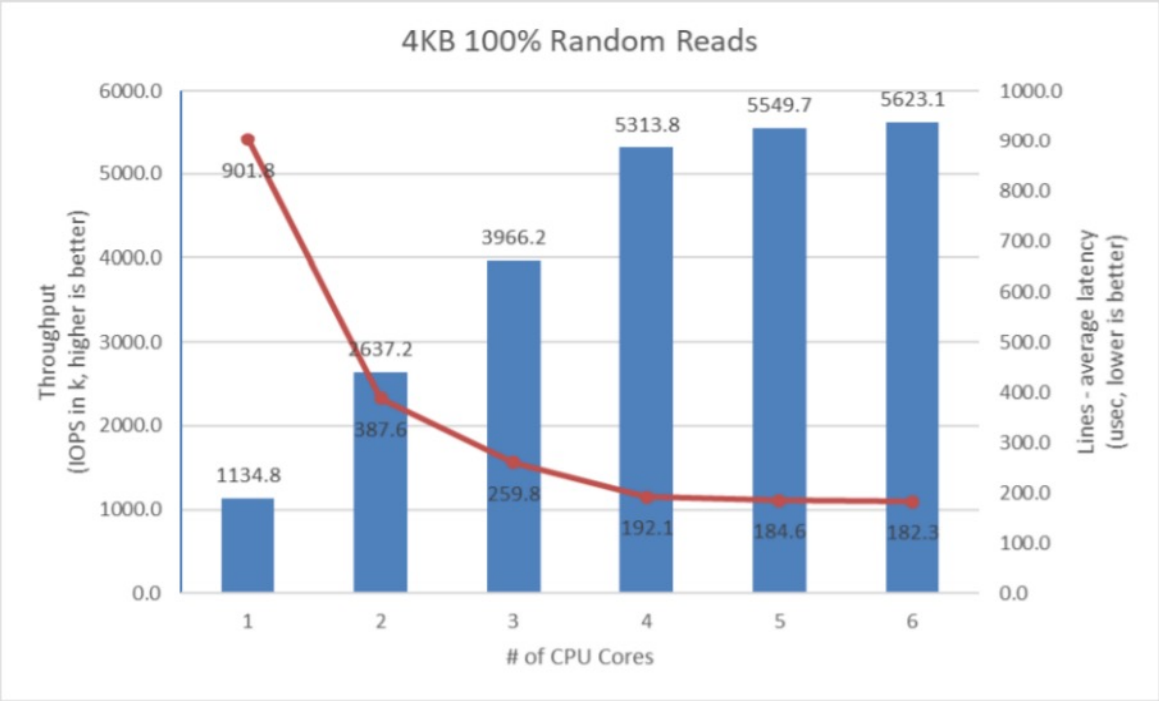
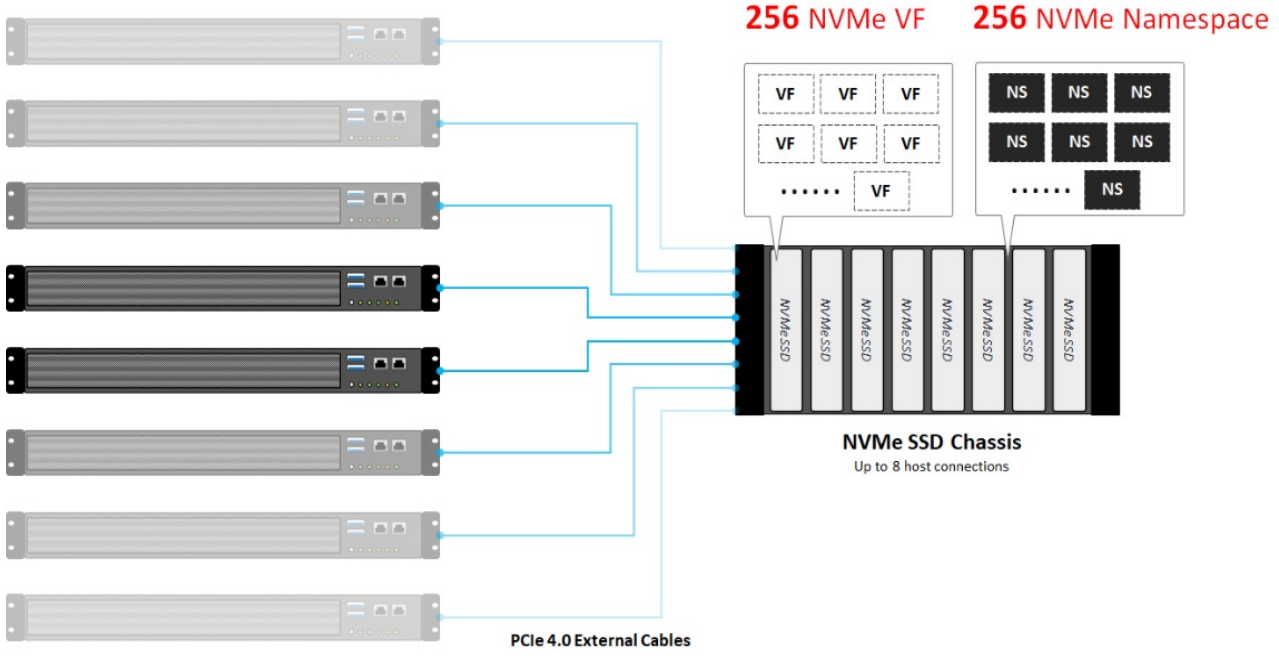


Figure 2: SPDK NVMe-oF RDMA Target I/O core scaling: IOPS vs. Latency while running 4KB 100% Random Read workload at QD = 64

Flexibility

- **Smaller grain & pooling**
  - Pooling 256 VFs and namespaces in one chassis. Enable precise storage allocation.
  - Connect to 8 hosts, adjust according to bandwidth requirement.



Implementation Challenges

Host side, Device side, SSD Management

Implementation Challenges

- **NVMe reset in OS and VM**
  - The mCPU should manage the reset requirement from the host OS and the virtual machine.

- **NVMe controller implementation**
  - Different vendors will have different ways or parameters to configure the virtual functions (controllers) and namespaces.

- **The PCIe resource limitation**
  - In some server BIOS, it might not be able allocate enough PCIe bus numbers or memory address under a specific PCIe bus.

Implementation Challenges

- **Admin command through out band**
  - Very limited support in administration commands through PCIe out of band.

- **NVMe sanitization before re-allocation (security delima)**
  - Data should be cleared before assigning to another hosts.
  - The mCPU should clear the data but the mCPU can see the data.

