SDC19 SNIA INDIA

May 23-24, 2019
Bangalore, India

STORAGE DEVELOPER CONFERENCE

NVMe-oF Ethernet SSD

Swati Chawdhary
Sandeep Kumar Ananthapalli

Samsung

Disclaimer

This presentation and/or accompanying oral statements by Samsung representatives collectively, the “Presentation” is intended to provide information concerning the SSD and memory industry and Samsung Electronics Co., Ltd. and certain affiliates (collectively, “Samsung”).While Samsung strives to provide information that is accurate and up-to-date, this Presentation may nonetheless contain inaccuracies or omissions. As a consequence, Samsung does not in any way guarantee the accuracy or completeness of the information provided in this Presentation.

This Presentation may include forward-looking statements, including, but not limited to, statements about any matter that is not a historical fact; statements regarding Samsung’s intentions, beliefs or current expectations concerning, among other things, market prospects, technological developments, growth, strategies, and the industry in which Samsung operates; and statements regarding products or features that are still in development. By their nature, forward-looking statements involve risks and uncertainties, because they relate to events and depend on circumstances that may or may not occur in the future. Samsung cautions you that forward looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements in this Presentation. In addition, even if such forward-looking statements are shown to be accurate, those developments may not be indicative of developments in future periods.

Agenda

- NVMe-oF protocol overview
- Native NVMe-oF prototype results
- Ethernet SSD and Use Cases
- Key Takeaways

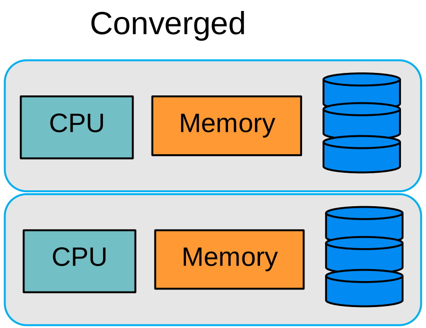
Hyperscale Datacenter and its Needs

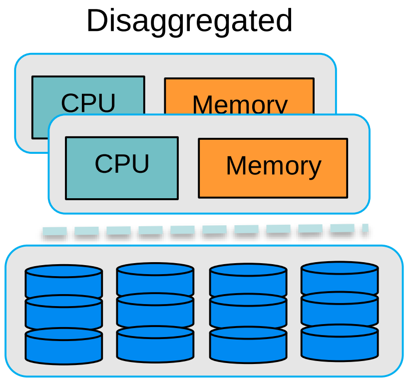
- Definition
  - Cloud operator that runs several hundreds of thousands of servers
  - E.g. Microsoft's Azure cloud service, Amazon web services, Google and Facebook

- Needs
  - Efficient infrastructure to provide services
  - Meet the growing needs of next gen applications e.g. AI, ML applications
  - Flexible, Scalable
  - Performance, power, cost-efficient

- Possible Solutions
  - **Disaggregation** of Storage and Compute
  - **Acceleration** of Compute : GPU, NIC, FPGA based storage accelerators

Why Disaggregation?





- Fixed ratio of Compute and Storage resources
- Resources scaled and managed together
- Upfront decision of resources for future needs
- Resources underutilized

- Compute and storage resources separated to create resource pools
- Resources scaled and managed independently
- Flexibility to size resources for different needs
- Improved resource utilization

SDC 19 SNIA INDIA

Disaggregation challenge

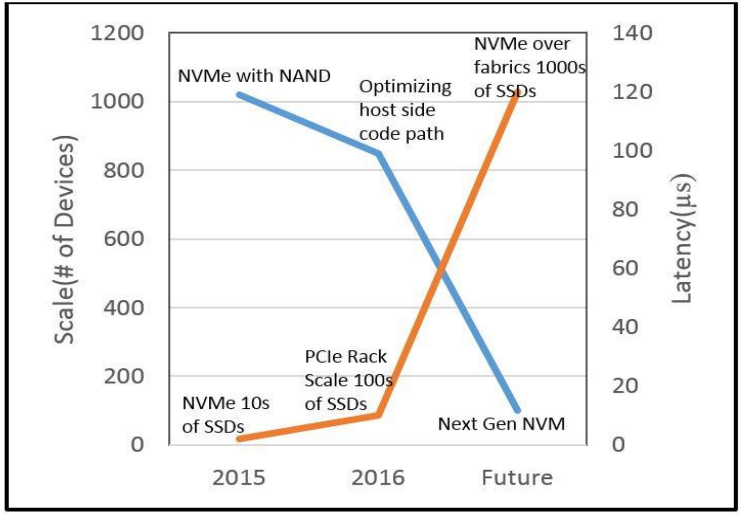
- Remote access overheads
  - Additional Interconnect latencies
  - Added Network and Protocol processing

- Disaggregation of Disk Storage(HDD) Common in Data Centers
  - Network overheads are small compared to HDD's millisecond access latency and low IOPs

- Disaggregation of NVMe Flash SSD is challenging
  - Network and protocol overheads are more pronounced compared to NVMe SSD's microsecond latency and high IOPS(~MIOPs)
  - Disaggregation of NVMe with iSCSI introduces performance drop

* "Flash storage disaggregation," EuroSys'16
* "NVMe-over-Fabrics Performance Characterization and the Path to Low-Overhead Flash Disaggregation", Systor'17

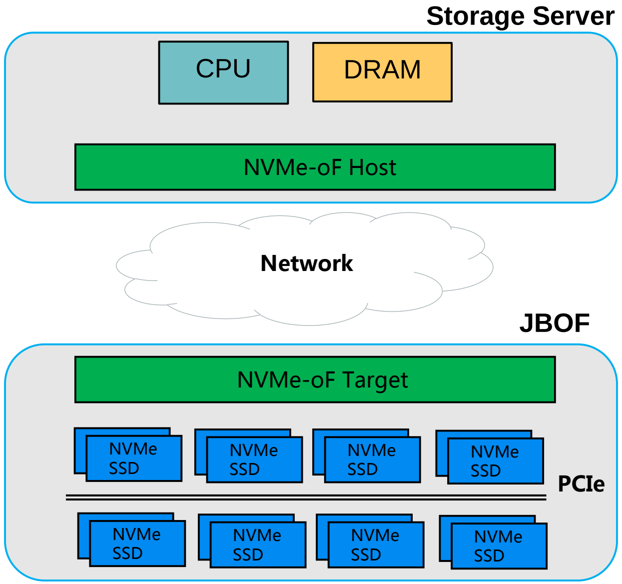
NVMe over Fabric Protocol



- Builds on top of NVMe Protocol and extends it to support various Network Transports(RDMA, FC, TCP)
- Enables scale-out of NVMe devices with DAS like performance and low latency(less than ~10µs)
- Avoids unnecessary protocol translation and offers an End-to-End NVMe model
- Architected from the ground up for current and Next-Generation NVM

SDC 19 SNIA INDIA

NVMe-oF JBOF

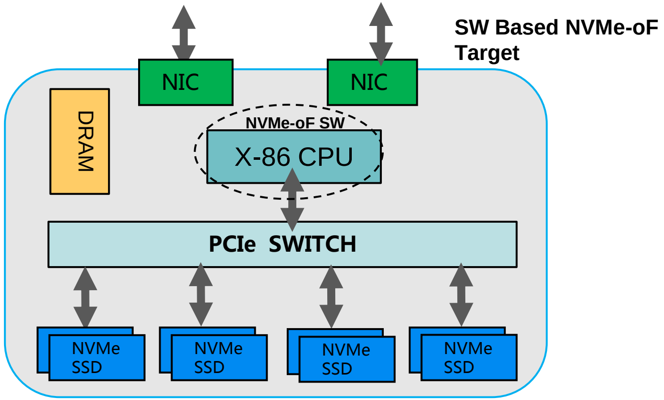


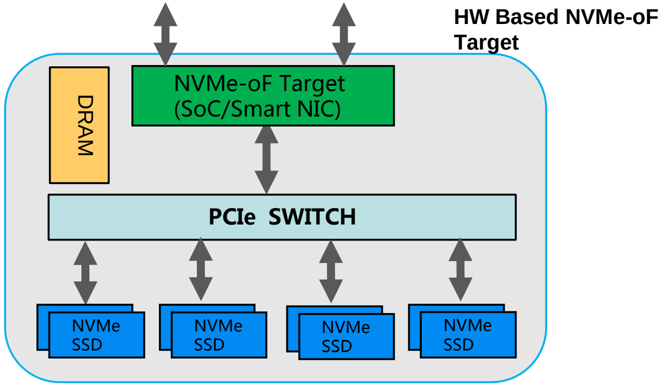
Disaggregated storage with NVMe-oF JBOF

- Enables disaggregation of NVMe
- End to End NVMe scale out
- Low latency
- Low cost
- High bandwidth
- High density

SDC 19 SNIA INDIA

Existing NVMe-oF JBOF Solutions

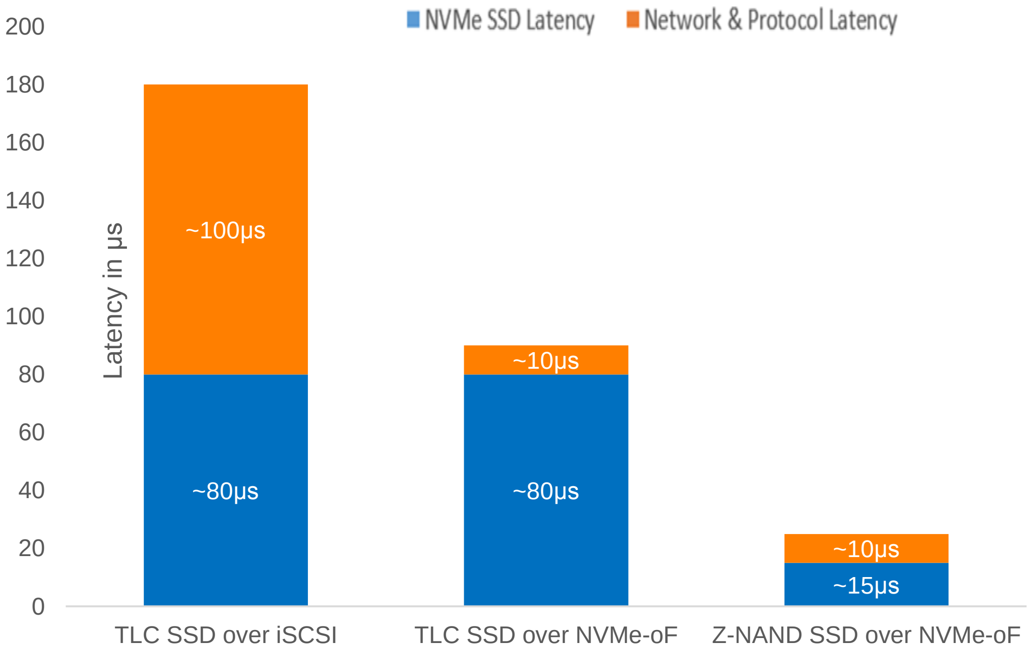




- NVMe-oF Target implemented either in Software or Hardware
- Require protocol conversion (NVMe-oF to NVMe)
- Latency offered meets NVMe-oF protocol goal of ~10µs and could vary based on the implementation type

SDC 19 SNIA INDIA

Remote access Latency Impact

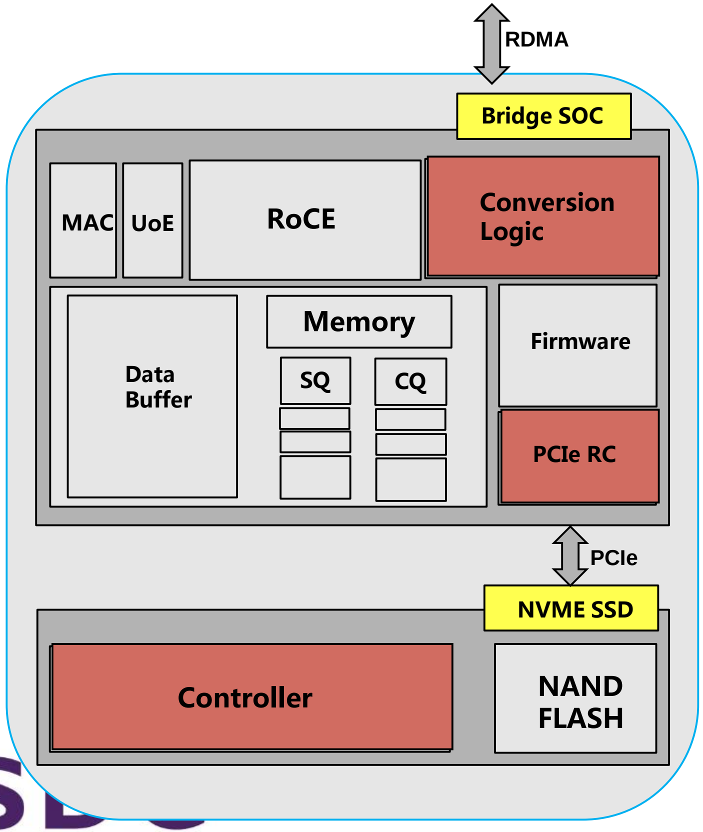


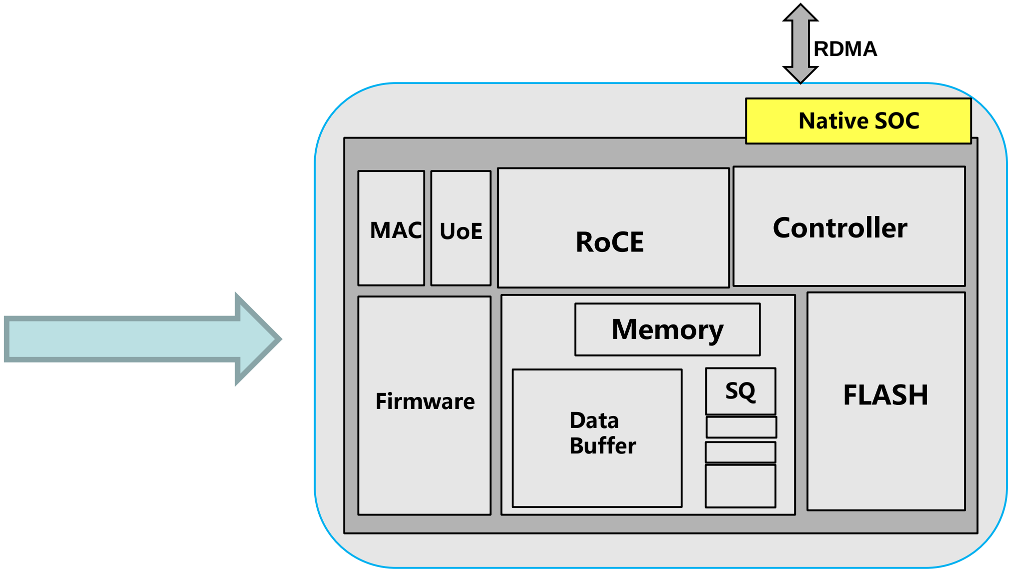
- Impact of Network and Protocol much more pronounced with Faster Storage like Z-NAND SSD
- Next Generation memory technologies may have read access latency under a microsecond
- Protocol implementation needs to be improved with **optimized design** for Faster Storage

Our Prototype Work and Results

Bridge NVMe-oF

Native NVMe-oF

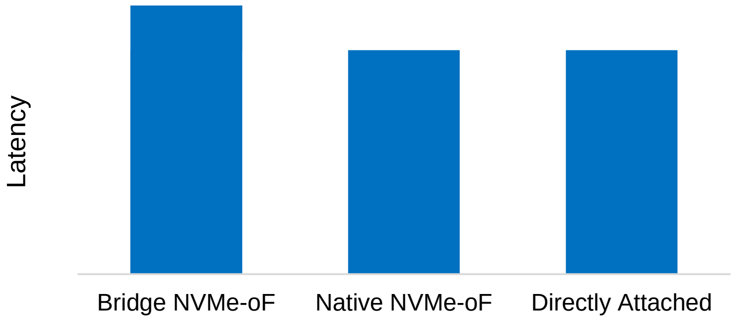


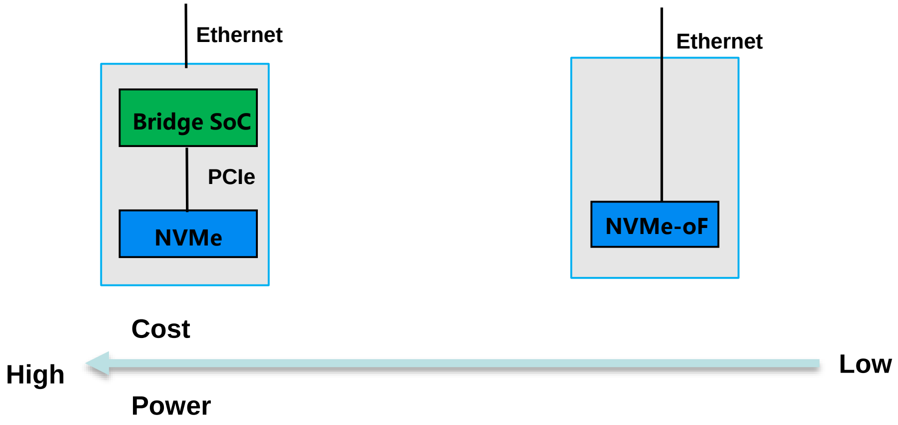


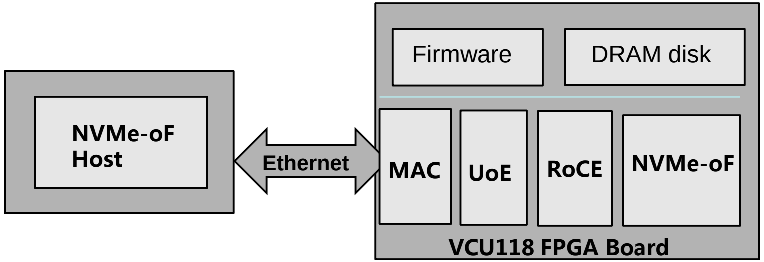
Experimental Results and Analysis

Bridge NVMe-oF : 20% more latency than DAS

Native NVMe-oF : Equivalent to DAS





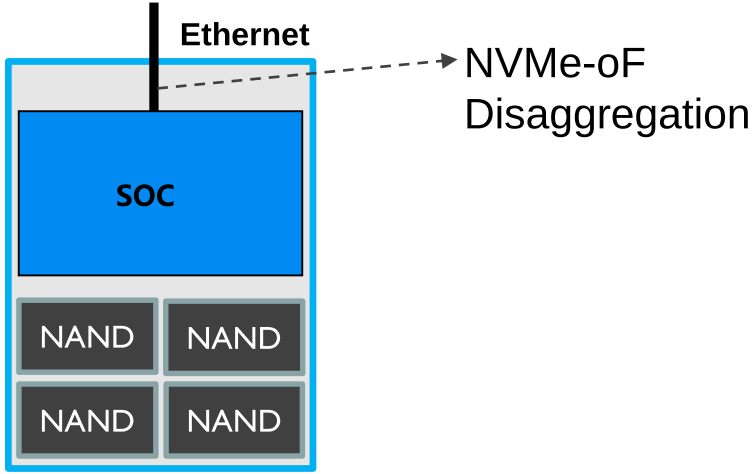


- Additional cost, power, footprint of the Bridge SoC
- Protocol Conversion
- Bridge SoC may become bottleneck for faster SSDs in the future

- Low cost, power, footprint due to single SoC
- No Protocol conversion
- DAS like performance and Latency

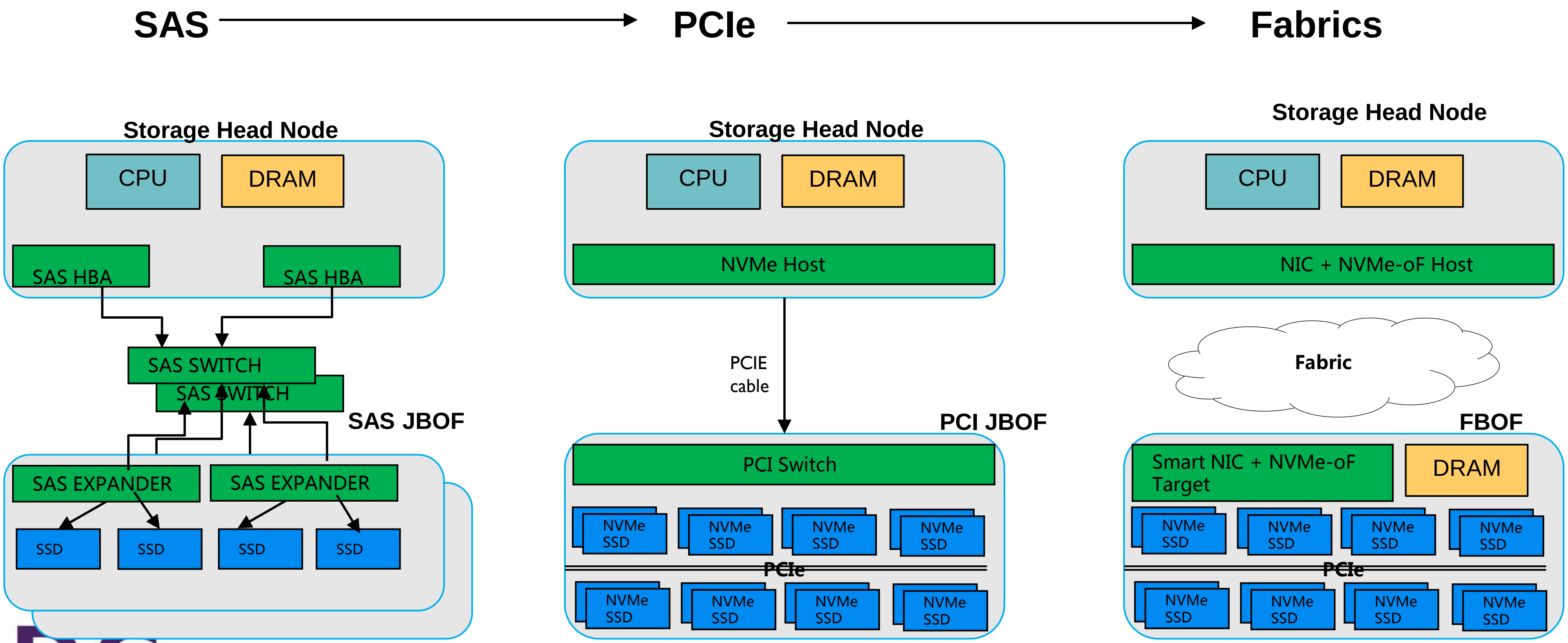
SDC 19 SNIA INDIA

Ethernet SSD for Disaggregation

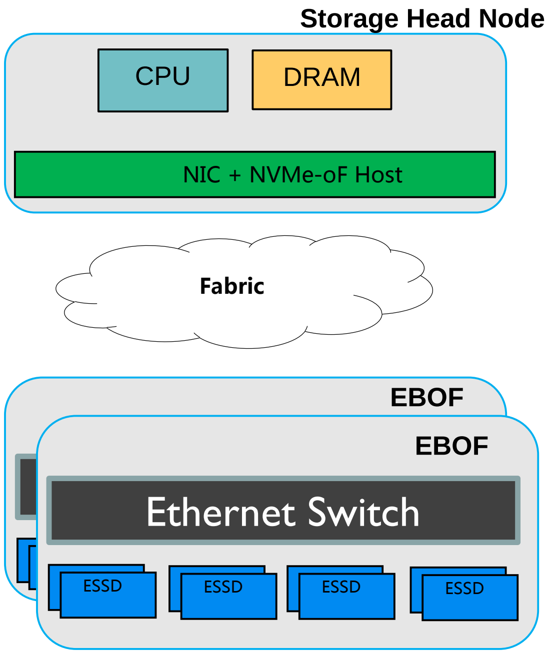


- A SSD with Ethernet port for storage drive level disaggregation
- Natively supports NVMe-oF protocol with an optimized design
- Integrates the fabric transport layer and the flash controller on a single SoC
- Low Latency, Power and Cost

Existing Solutions

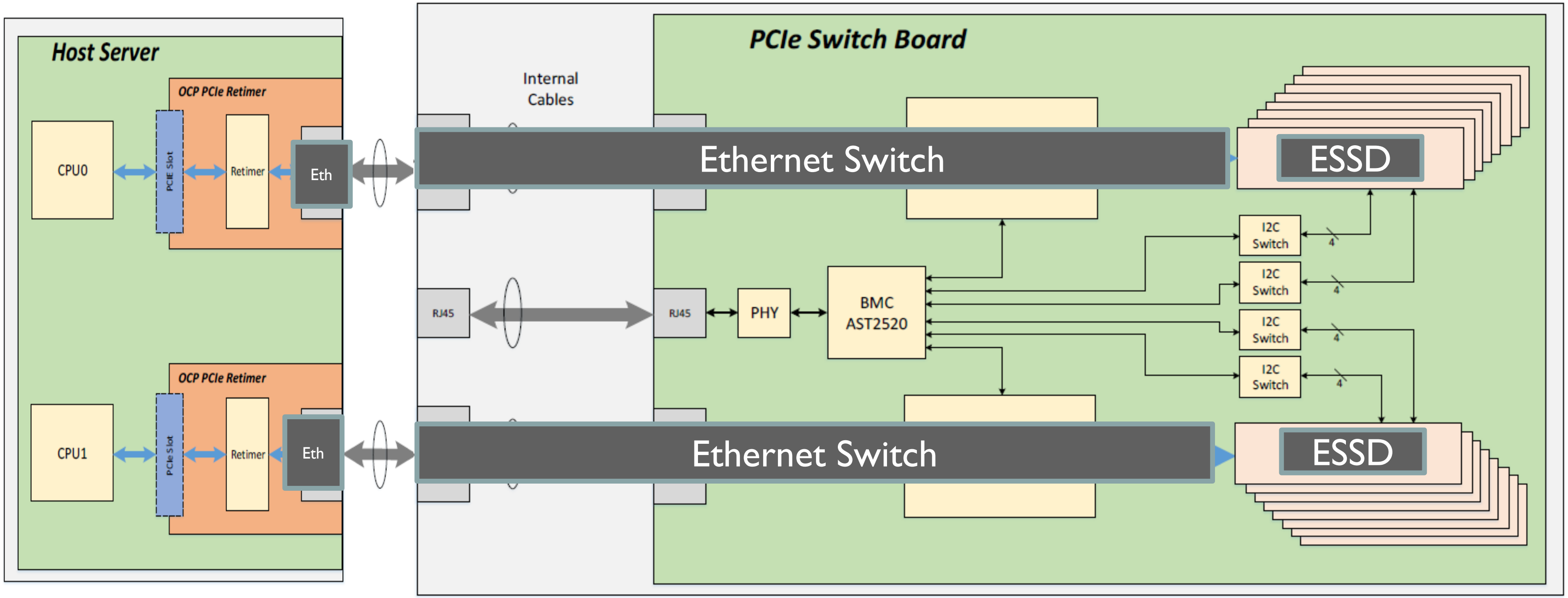


Ethernet SSD based EBOF



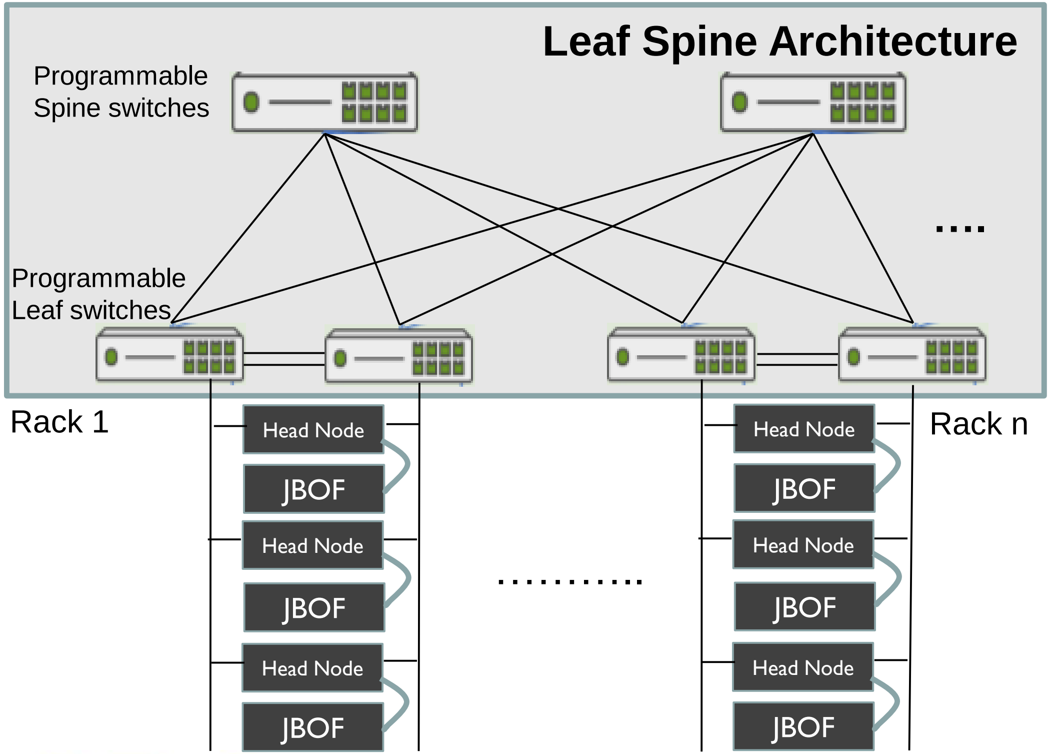
- Pros
  - A simple backplane design that offers **<u>higher density and full utilization of flash</u>** attached
  - Scalable(Ethernet), Shareable(NVMe-oF), Extendable(Daisy chaining), Efficient(Low latency)
  - Lesser power because flash natively talks to Ethernet
- Cons
  - Ecosystem to be re-architectured to manage many network devices

OCP FX-16: Proposed changes



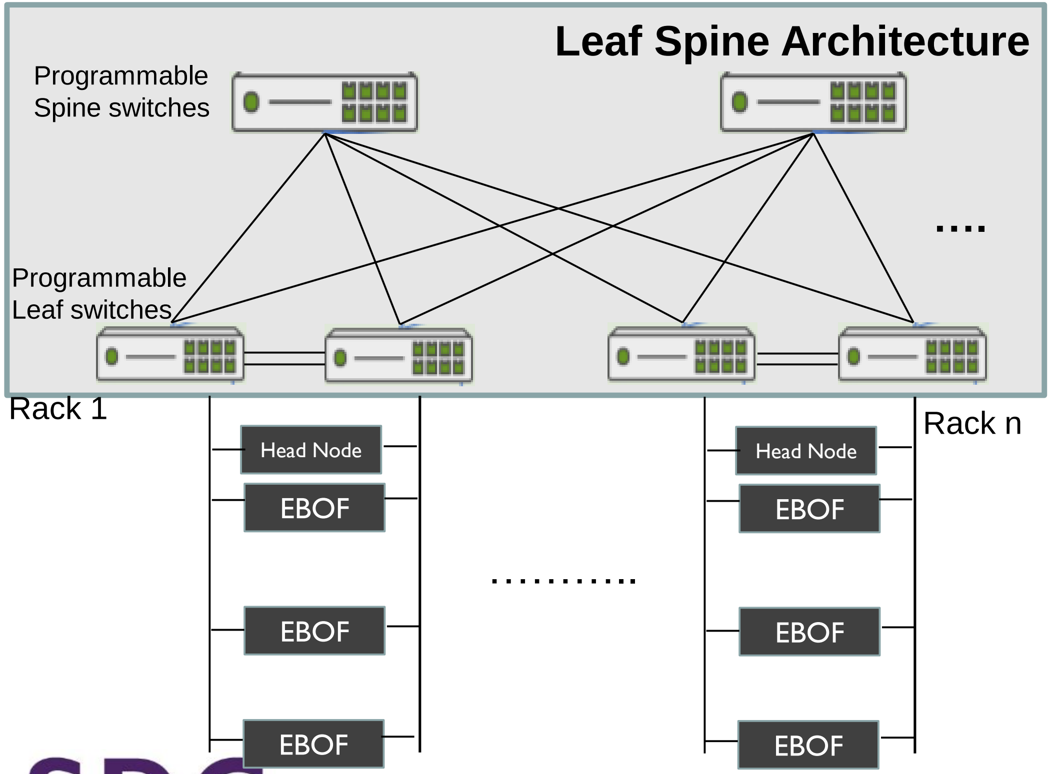
* Ethernet as The Next storage-Fabric of Choice – OCP summit 2019

Deployment of JBOF



- Current
  - Today many JBOF solutions are deployed in leaf-spine architecture
- Pros
  - Easily deployed, managed, monitored, serviced in current datacenter architecture
- Cons
  - Scalability limitation of PCIe switch
  - Extra Head node for each additional JBOF

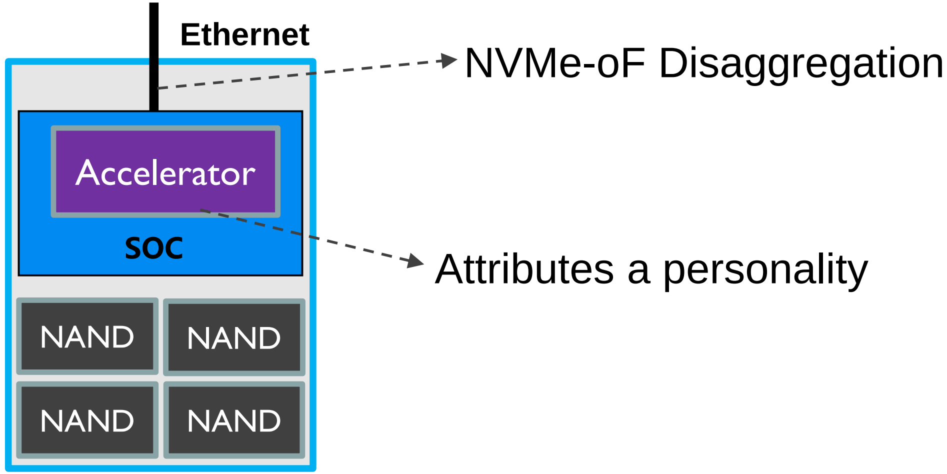
Deployment of EBOF



- Future
  - See EBOF getting deployed in future
- Pros
  - Add just the EBOF on demand for more capacity, no need for deploying additional Head node
  - Savings on TCO on **Head Node, Power, Rack Space**
- Cons
  - Requires a new management system to handle many ESSDs
  - New network architecture to reduce switching cost

SDC 19 SNIA INDIA

Ethernet SSD for Disaggregation and Acceleration

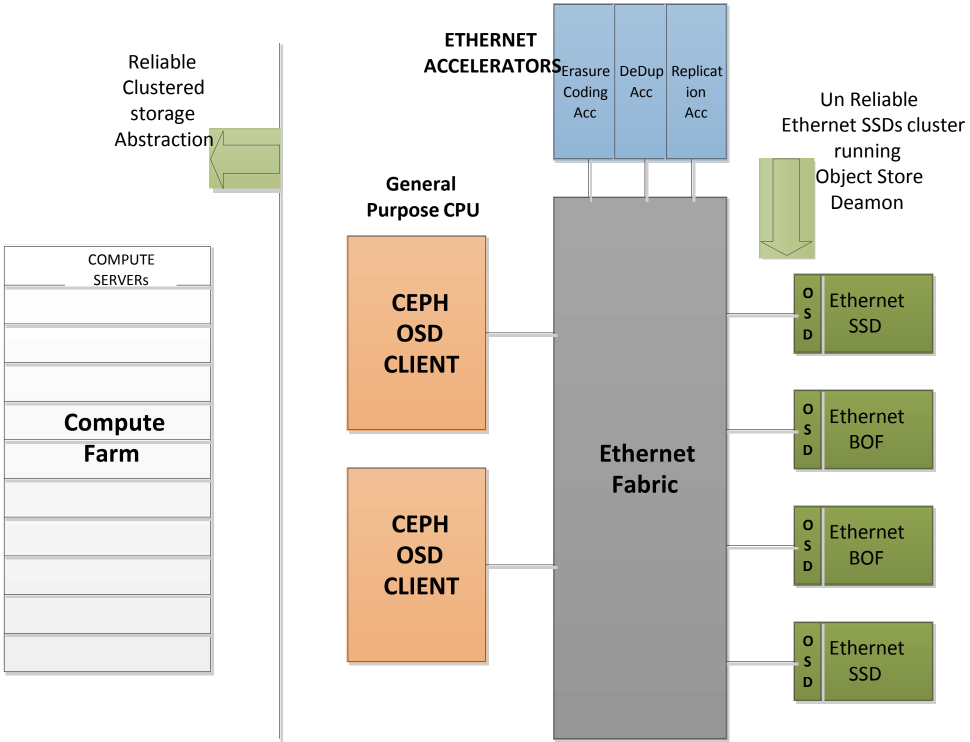


- On-chip acceleration attributes an application specific personality to the Ethernet SSD
- CPU Offload by use of Accelerator
- Enables new use cases in the area of Object storage, CDN, IOT etc.

SDC 19 SNIA INDIA

NVMe-oF Ethernet SSD + Acceleration based Use Cases

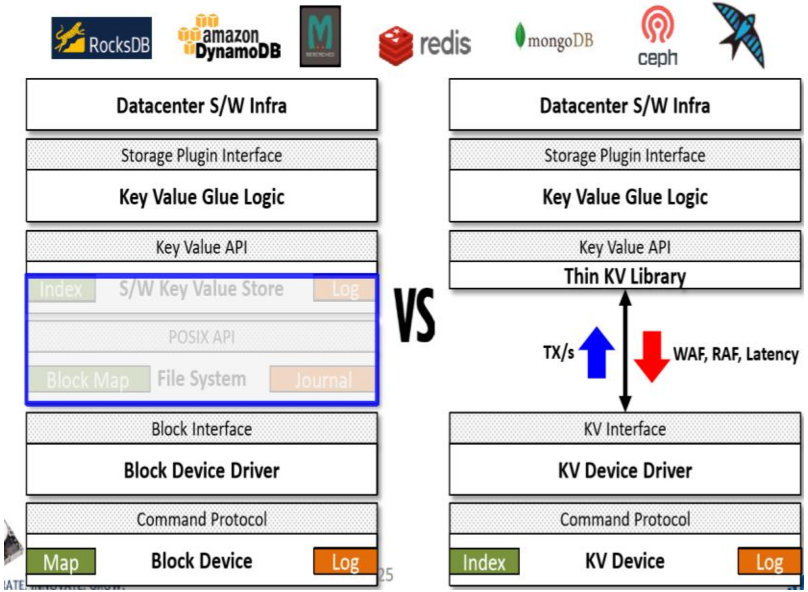
Big Data: Object Storage



- Personality
  - Object Storage Drive
- Need
  - Large Scale Reliable Unified Object Storage
- Solution
  - Replace traditional storage nodes with ESSD/EBOF running OSD daemons
  - Use Ethernet accelerators for RAID, Deduplication, Replication, Compression for added value
- Benefit
  - Cheaper per GB CEPH storage
  - More density in a rack

SDC 19 SNIA INDIA

Big Data: Key/Value Storage



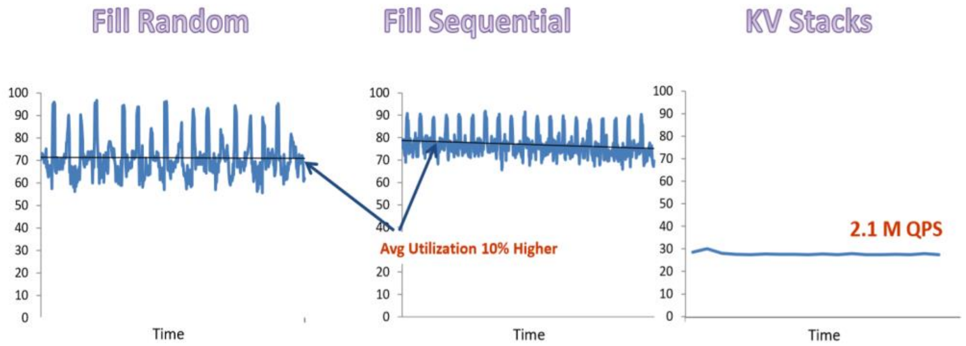
Current Block based stack

Optimized KV stack

- Personality
  - KV Store SSD
- Need
  - Efficient and optimized KV store
- Solution
  - Store KV objects natively into KV ESSD
- Benefit
  - Storage node is offloaded with file-system abstraction
  - Exact amount of User I/O data is written and no extra system writes to disk

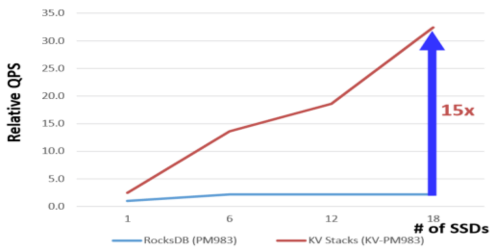
SDC 19 SNIA INDIA

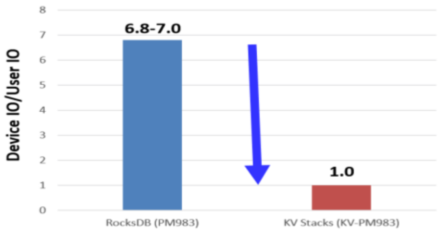
Big Data: Key/Value Storage



Avg 170K QPS@72% CPU | Avg 400K QPS@80% CPU | Avg 2.1M QPS@30% CPU

- **15x IO performance over S/W key value store on block devices**





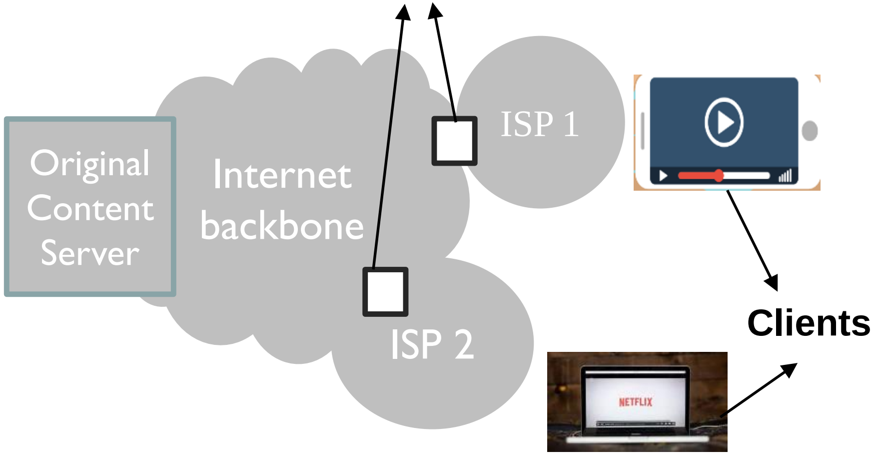
- Setup
  - 4 KV clients talking to 18 KV SSDs in a PCIe based back plane
- Observation
  - With block based host stack + block based SSD, CPU saturates much faster; whereas optimized KV stack+ KV SSD can get much better throughput with only 30% CPU utilization
  - If application needs more capacity and throughput, add one more EBOF and offer **2x throughput and capacity** without needing to add one more head node

* https://www.snia.org/sites/default/files/SDC/2017/presentations/Object_ObjectDriveStorage/
Ki_Yang_Seok_Key_Value_SSD_Explained_Concept_Device_System_and_Standard.pdf

SDC 19 SNIA INDIA

Edge Caching for CDN

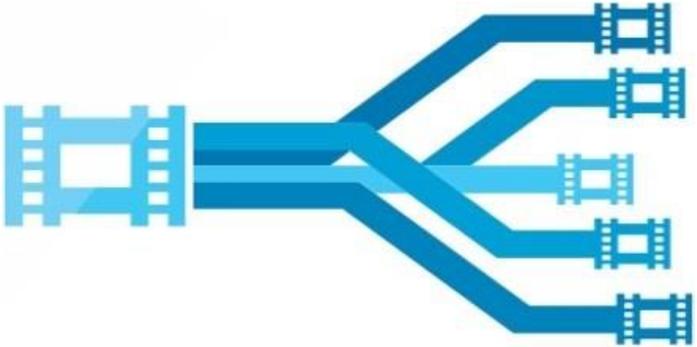
Replace traditional Edge servers with ESSD



- Personality
  - Edge Caching SSD
- Need
  - Minimize the streaming media traffic to original content server
- Solution
  - Best effort delivery of popular content from Edge servers
  - Ethernet SSD - to **natively sniff** ethernet traffic. Additional logic for caching decision inside ESSD can potentially replace traditional Edge servers
- Benefit
  - Lesser cost, power, space for Edge infrastructure

SDC 19 SNIA INDIA

Streaming Media Transcoding



- Personality
  - Media transcoder SSD
- Need
  - Faster streaming require efficient transcoding
- Solution
  - Use transcoding accelerator inside ESSD
  - To offload the server and deliver the transcoded bit stream to server
- Benefit
  - Media server can issue more i/o requests to EBOF for faster streaming

SDC 19 SNIA INDIA

Key Takeaways

- Hyperscale datacenters are moving towards a disaggregated, accelerated environment
- NVMe-oF protocol enables disaggregation of NVMe devices without any additional overheads
- Native NVMe-oF design removes the latency overheads of Bridge NVMe-oF design
- A Native NVMe-oF Ethernet SSD enables an ecosystem with better connectivity, disaggregated storage, scalability, throughput, cost
- EBOF fits homogeneously in a leaf spine based switching architecture and saves on extra head node cost for additional capacity
- NVMe-oF App SSD can open various new use cases

References

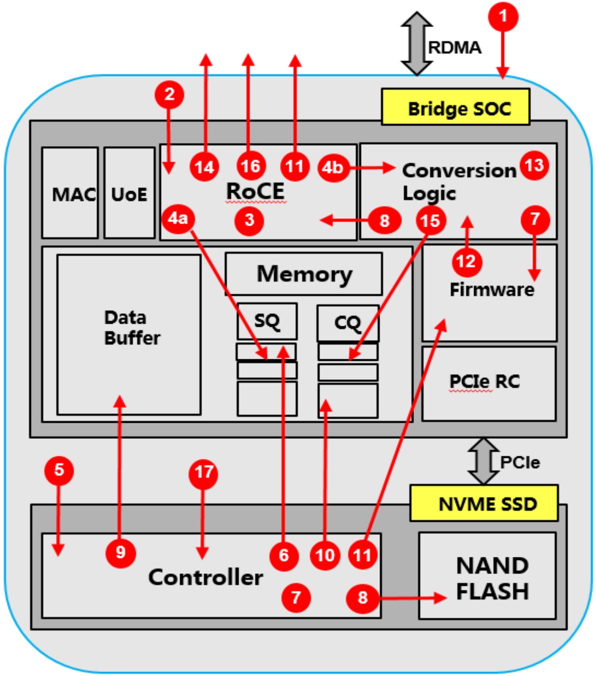
- https://nvmexpress.org/wp-content/uploads/NVMe_Over_Fabrics.pdf
- http://csl.stanford.edu/~christos/publications/2016.flash.eurosys.pdf
- https://pdos.csail.mit.edu/6.824/papers/borg.pdf
- https://www.samsung.com/us/labs/pdfs/nvmf-disaggregation-preprint.pdf
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8071053
- https://www.snia.org/sites/default/files/CSI/Why-Composable-Infrastructure-Final.pdf
- https://www.cloudflare.com/learning/cdn/glossary/edge-server/
- http://iran-lms.com/images/images/Books/PDF/Cloud-Computing-Cambridge-University-Press-2017-Sandeep-Bhowmik.pdf
- Software Defined Fabric for OCP-based Leaf & Spine Switches – OCP 2019
- Project Zipline by Microsoft https://github.com/opencomputeproject/Project-Zipline

SDC 19 SNIA INDIA

THANK YOU!

Contact: s.chawdhary@samsung.com
a.sandeep@samsung.com

Bridge NVMe-oF – Read Flow



Command submission

1. Host issues NVMe-oF Read Command
2. RoCE receives NVMe-oF Read Command encapsulated in RDMA Send packet
3. RoCE reserves Data Buffers and prepares PRP/SGLs

4a. RoCE updates the SQ entry in Submission Queue

4b. RoCE sends the command info to Conversion logic

5. Conversion logic rings the doorbell

Command processing

6. Controller fetches the command from submission queue
7. Controller processes the Read command
8. Controller fetches the data from NAND Flash
9. Controller writes the data into Data Buffer

Controller posts the CQ entry in Completion Queue

Controller Posts Interrupt

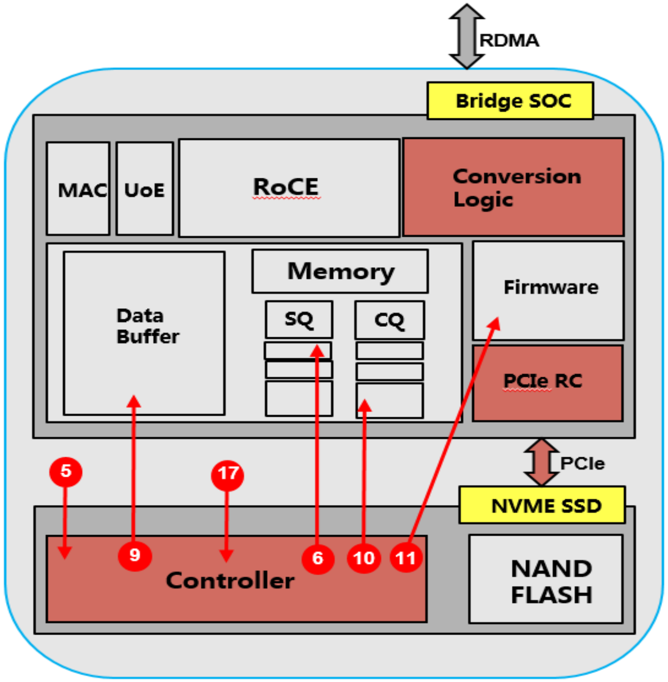
12. Firmware triggers conversion logic to read data from Data buffer
13. Conversion logic converts Read data as payload for RDMA Write
14. RoCE transmits RDMA Write packet to Host

Command completion

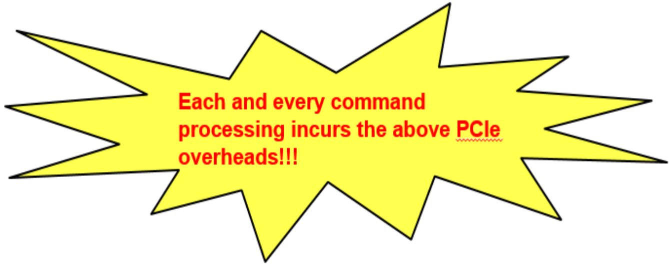
15. Conversion logic converts CQ entry in Completion Queue to RDMA send Packet
16. RoCE Transmits NVMe completion encapsulated in RDMA Send Packet
17. Conversion logic updates the Completion Queue head doorbell

SDC 19 SNIA INDIA

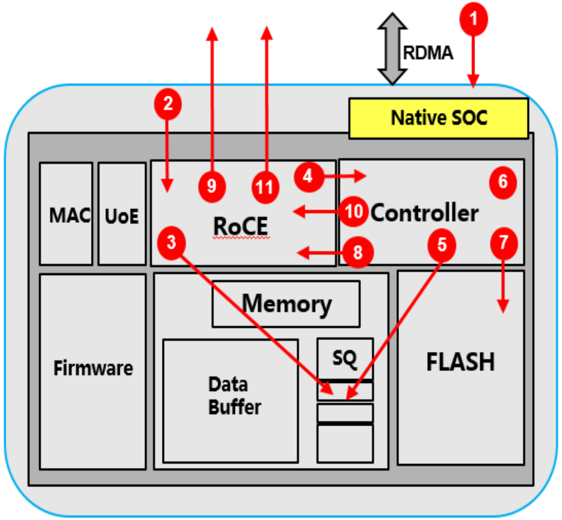
Bridge NVMe-oF – Overhead



- PCIe memory write for doorbell ring (step 5)
- PCIe memory read for command fetch (step 6)
- PCIe memory write for data buffer update (step 9)
- PCIe memory write for Completion posting (step 10)
- PCIe memory write for MSIX posting (step 11)
- PCIe memory write for doorbell ring (step 17)



Native NVMe-oF – Read Flow



Command submission

1. Host issues NVMe-oF Read Command
2. RoCE receives NVMe-oF Read Command encapsulated in RDMA send Command
3. RoCE posts SQ entry in Submission Queue
4. RoCE notifies the controller about the arrival of new SQ entry

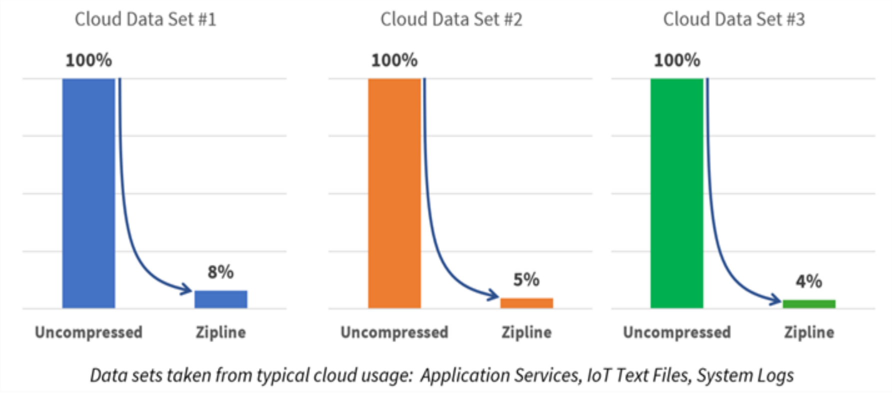
Command processing

5. Controller Fetches the command from submission queue
6. Controller processes the Read command
7. Controller Fetches data from Flash
8. Controller Sends data to RoCE
9. RoCE encapsulates Read data as payload for RDMA Write and sends to Host

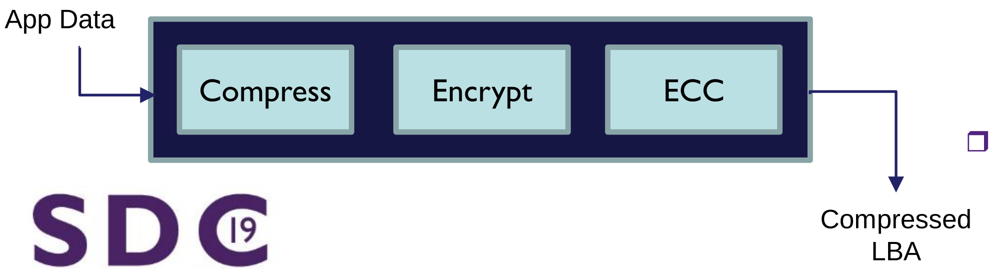
Command completion

10. Controller posts the CQ entry to RoCE
11. RoCE encapsulates CQ entry into a RDMA Send packet and sends to Host

BIG Data: Compression



Project Zip-line



- Personality
  - SSD with Compression capability
- Need
  - Manage huge volumes of data efficiently
- Solution
  - Use compression accelerator inside ESSD (similar to Project zip-line)
  - Do encryption, ECC after compression
  - Compression engines add almost no overhead on Flash write operation
- Benefit
  - 90% lesser writes to flash.
  - Improved flash endurance and write IOPS