Next-Gen NVMe-oF Reference System: From Media to Network

2020. 11.

Duckho Bae

Samsung Electronics

Agenda

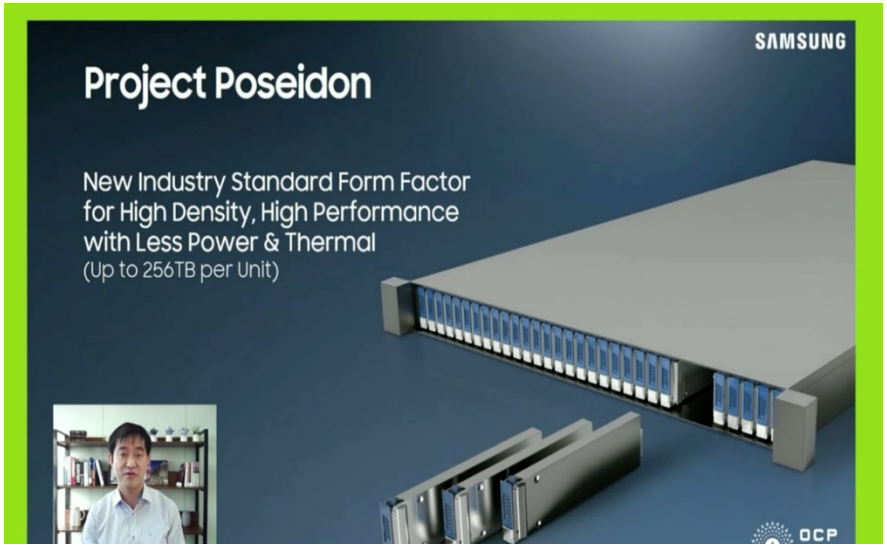
- Why Open-Source Reference for NVMe-oF System?
- Software Solution for NVMe-oF
- Future Works & Conclusion

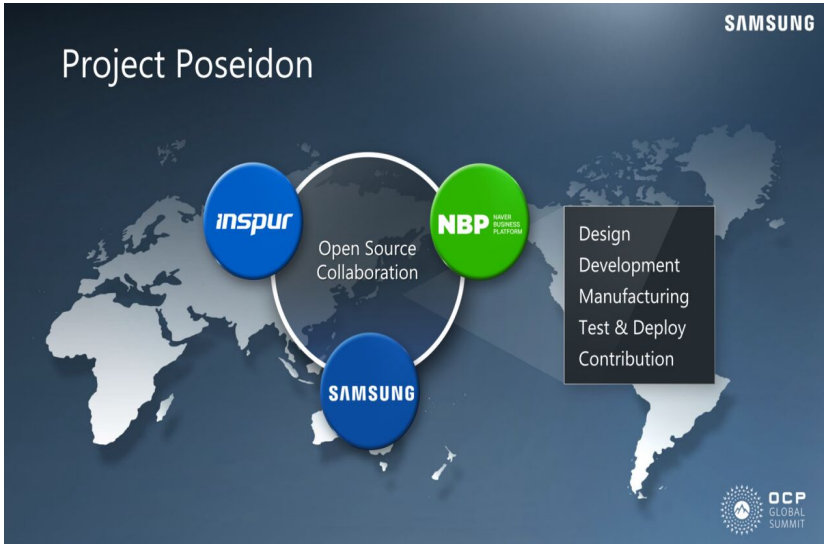
소개

- **2020 – Current: Principal Engineer, Memory Division, Samsung Electronics**
- 2013 – 2019: Staff Engineer, Memory Division, Samsung Electronics
- 2008 – 2013: Ph.D., Hanyang University
- 2006 – 2008: M.S., Hanyang University
- 2002 – 2006: B.S., Hanyang University

Project Poseidon

- **Announced in 2020 OCP Global Summit**
- **Open reference storage platform based on 3-way collaboration**

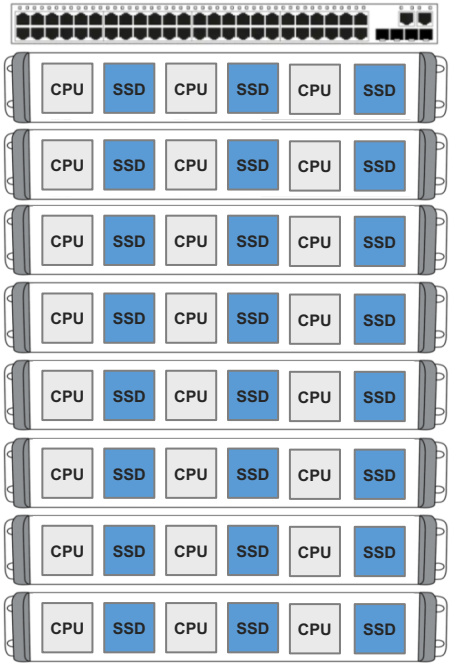




Disaggregated Architecture

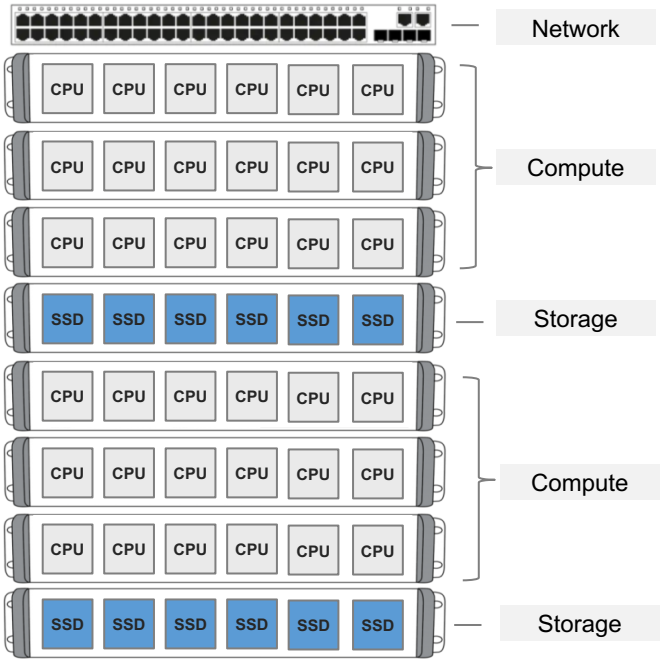
- Storage system is evolving towards the disaggregated architecture

Direct Attached Architecture



- Increase CPU & SSD utilization
- Reduce storage spending & TCO
- Simplified scalability
- Higher performance
- Increase hardware flexibility
- Ideal capacity utilization

Disaggregated Architecture

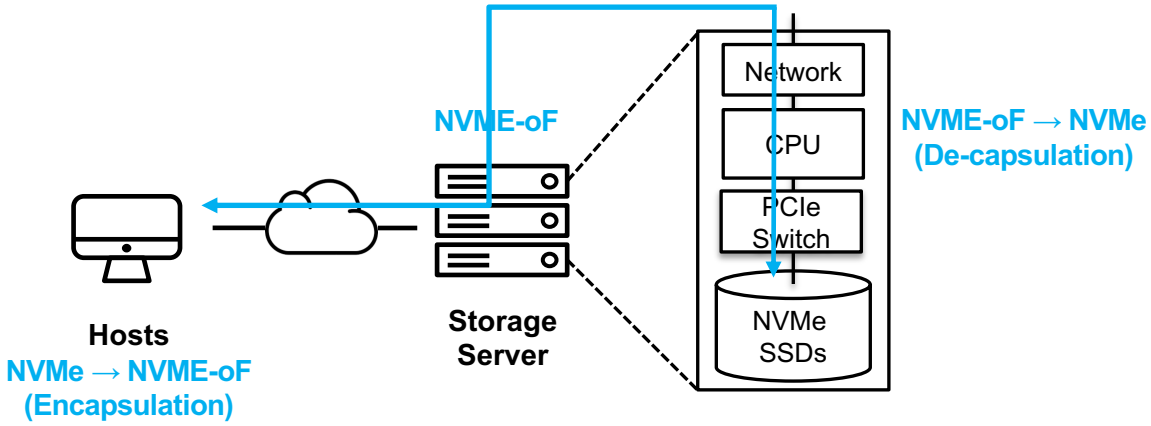


NVMe-oF Interface

- **Can break through the scaling limitation of PCIe-attached NVMe**

※ Up to few hundreds

- **Uses a transport protocol over a network to access remote NVMe**
  - End-to-End NVMe semantics across a range of topologies
  - Retains NVMe efficiency and performance over network fabrics



Why NVMe-oF ‘Reference’ System

- **NVMe ecosystem is expanding rapidly**
  - NVMe grows 61.4% share by 2020 in enterprise storage * Source: IDC
- **Storage disaggregation has become major trend in datacenter**
  - Can scale storage resources independently in a cost effective and flexible manner
- **Next-generation storage brings strict requirements**

  * PCIe Gen4/Gen5, CXL, E1.x, E3.x, …
  - More power, higher density, higher throughput, finer QoS control are required
- **There is few 'open-sourced' reference system for NVMe-oF**
  - To leverage NVMe-oF eco system

>> Both HW and SW open-sourced references for NVMe-oF are needed!

Software Solution for NVMe-oF

Prerequisite for Software (1/3)

1. Abstracts logical volumes from physical devices
2. Supports various types of transport bindings
3. Fully utilize the performance of NVMe SSD
4. Provides flexible NVM subsystems

Prerequisite for Software (2/3)

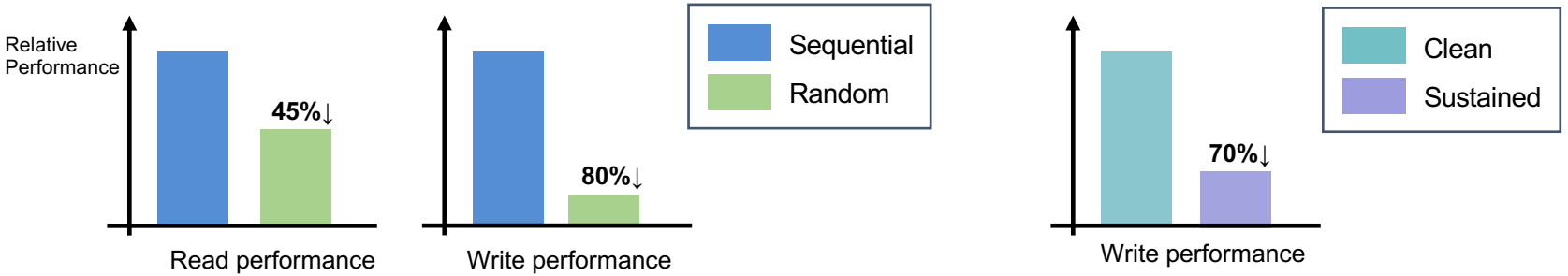
1. **Abstracts logical volumes from physical devices**
   - Make physical devices invisible to the initiators
   - Add storage intelligence features – Volume manage, RAID, compression, tiering, …
     - Requires metadata like mapping table management

2. **Supports various types of network protocols - RDMA, FC, TCP**
   - NVMe/TCP by default (ratified on Nov. 2018)
     - Enables datacenters to utilize their existing TCP/IP network
     - Offers tens of us latencies (normally, 40us ~ 90us)
     - Ready for future hardware-accelerated implementations

Prerequisite for Software (3/3)

3. Fully utilize the performance of NVMe SSDs

- Sequential performance of NVMe SSDs is much faster than random IO
- Clean state performance in write is superior than sustained performance



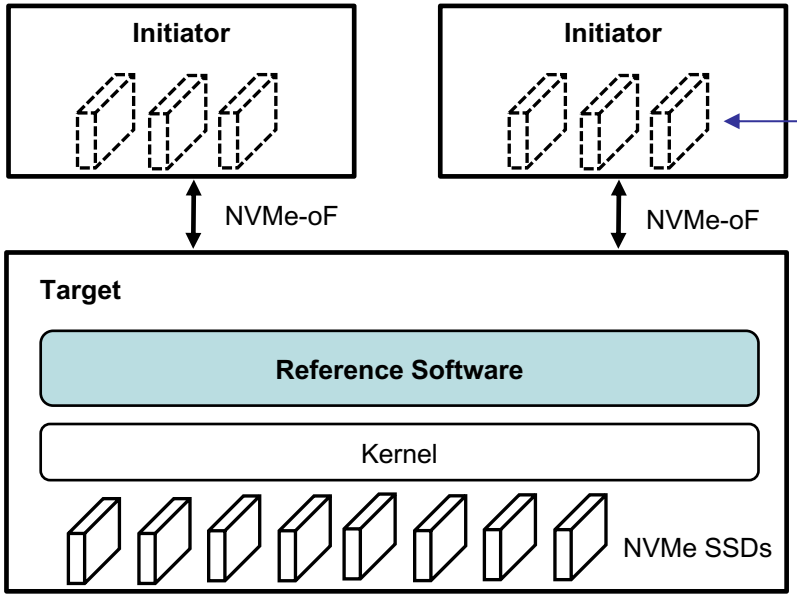
※ Samsung PM9A3 PCIe Gen4 NVMe SSD

- Need to understand the characteristics and limitations of NAND Flash
  - Ex. Different program/erase units, not allows in-place updates, EPI, read reclaim, …

※ Erase program interval

Software Concept

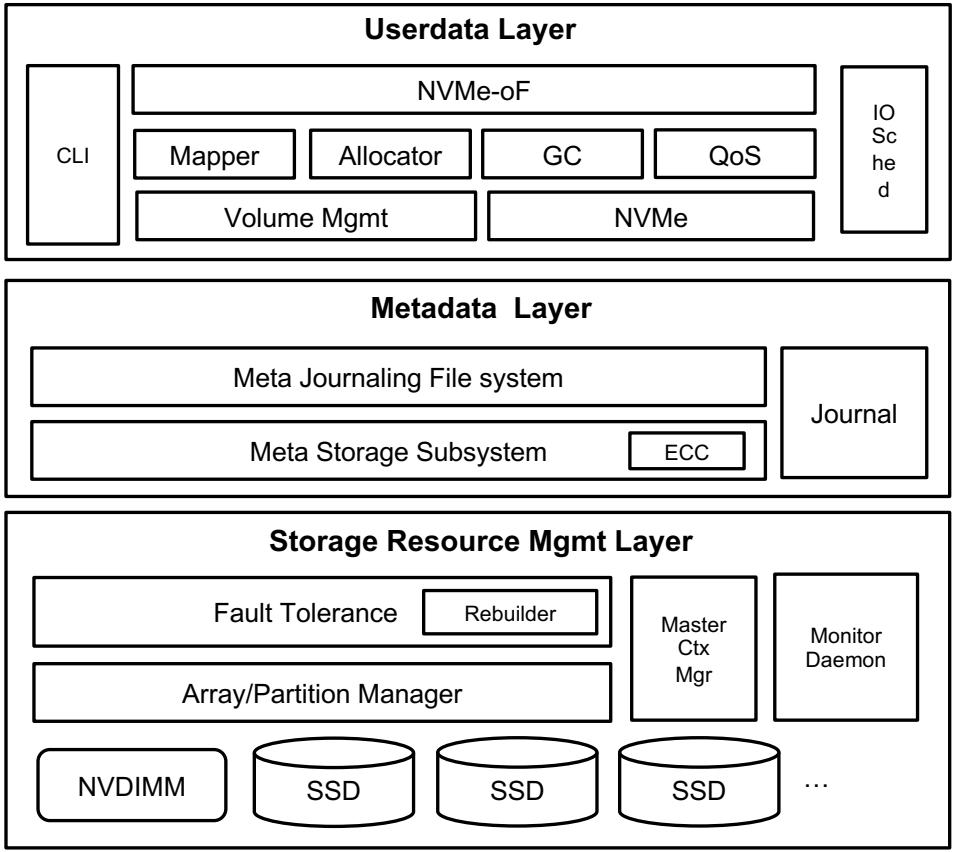
- **Runs as user application**
- **Provide ‘customized’ virtual devices to initiators via NVMe-oF interface**



Customized virtual device

※ Example of customized options for each virtual SSDs
- Capacity
- Performance (IOPS, BW, QoS)
- Features
  - RAID (1, 5, 6, …)
  - Compression
- DWPD
- Energy consumption
- … and MORE!

Software Stack



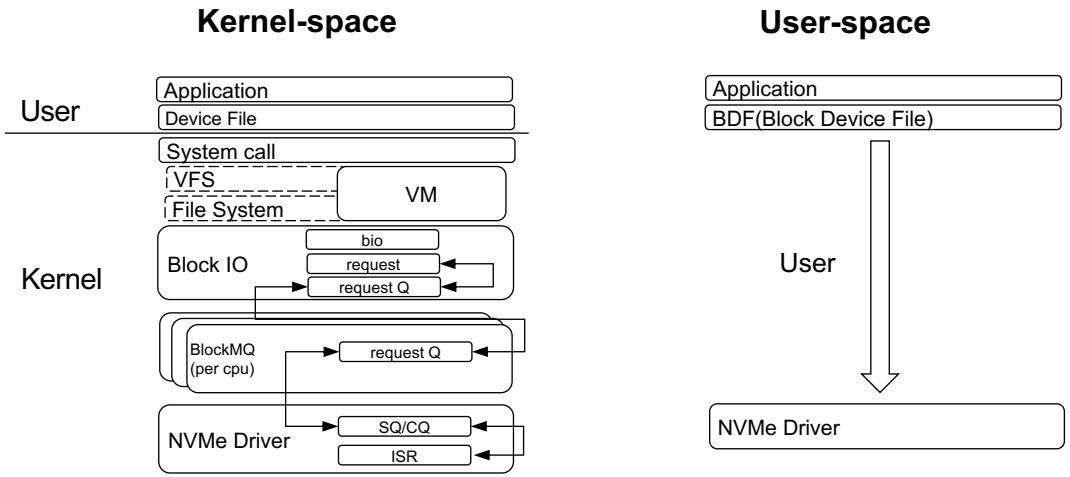
- User I/O handling
- Provides NVMe-oF connectivity
- Logical Device mgmt - Volume
- Performance Optimization
- User IO QoS

- Guarantee ACID of metadata
- Metadata I/O handling
- Journaling and Restore

- Provides Fault Tolerance Feature - RAID / EC
- Partition mgmt - System / User / Meta Area
- Physical device mgmt - SSD Array, NVDIMM
- SSD device monitoring

Characteristics of Software (1/2)

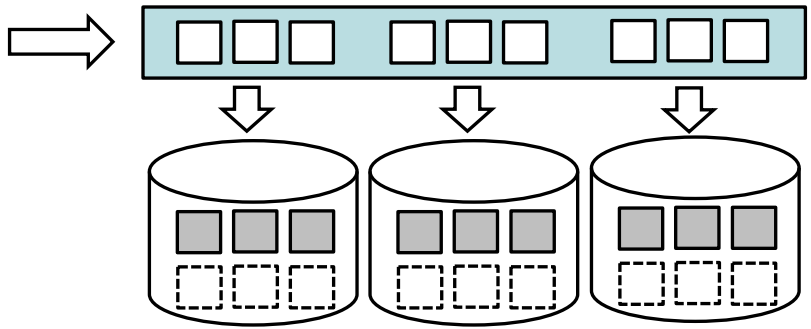
- **User-space NVMe-oF / NVMe IO**
  - Avoids overheads of system calls and data copies
  - Spends more CPU cycles for storage services
  - Enables better latency and IOPS



Characteristics of Software (2/2)

- **Write buffer makes SSD-friendly writes IO**
  - Can shorten write latency and make QoS stable if write buffer is placed in NV memory
  - Can enjoy the sequential write performance of NVMe SSDs
  - Can make SSDs to clean state using TRIM command

① All user writes are first stored in write buffer



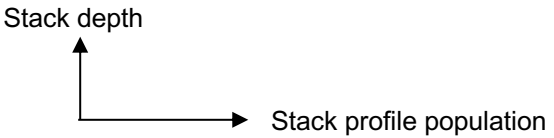
② When write buffer is full, flush user data in log-structured manner

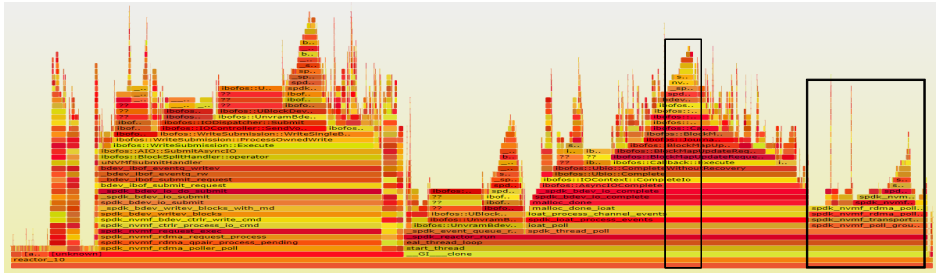
Challenges & Approaches

- **Challenge 1: Saturate high-bandwidth network in TCP**
- **Challenge 2: Initiator SW stack becomes more important**
- **Challenge 3: Efficient internal metadata management**
- **Challenge 4: Providing fault tolerance when NVMe drive fails**

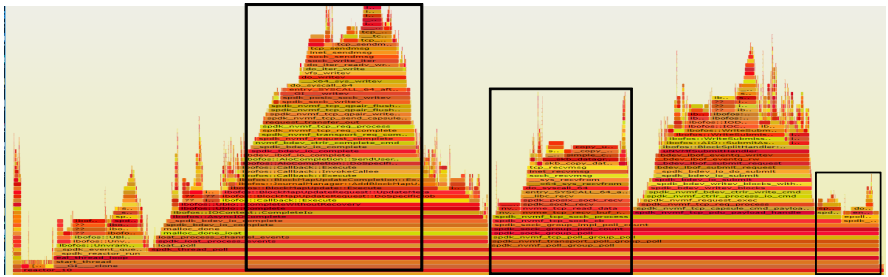
Challenge 1: Saturate high-bandwidth network in TCP

- **Basically, follows *SPDK* philosophy**
- **Even if we use SPDK, CPU is still bottleneck**
  - Harsh with small IO (4KB)
  - TCP stack makes worse!





RDMA (4KB writes)

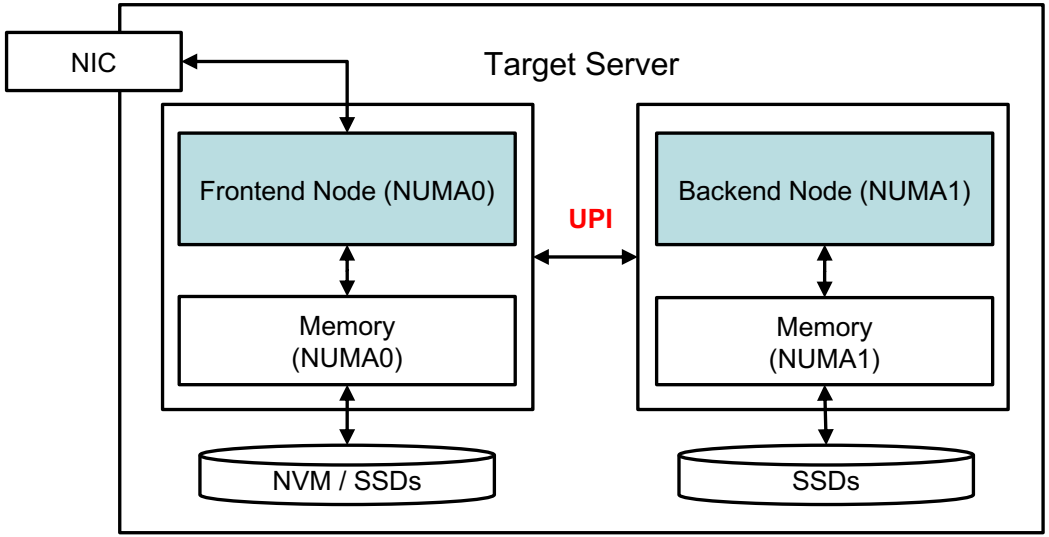


TCP (4KB writes)

- **TCP consumes 3 times more CPU resources than RDMA! (35% vs 12%)**
- **In PCIe Gen4, this would be much worse!**

Challenge 1: Saturate high-bandwidth network in TCP

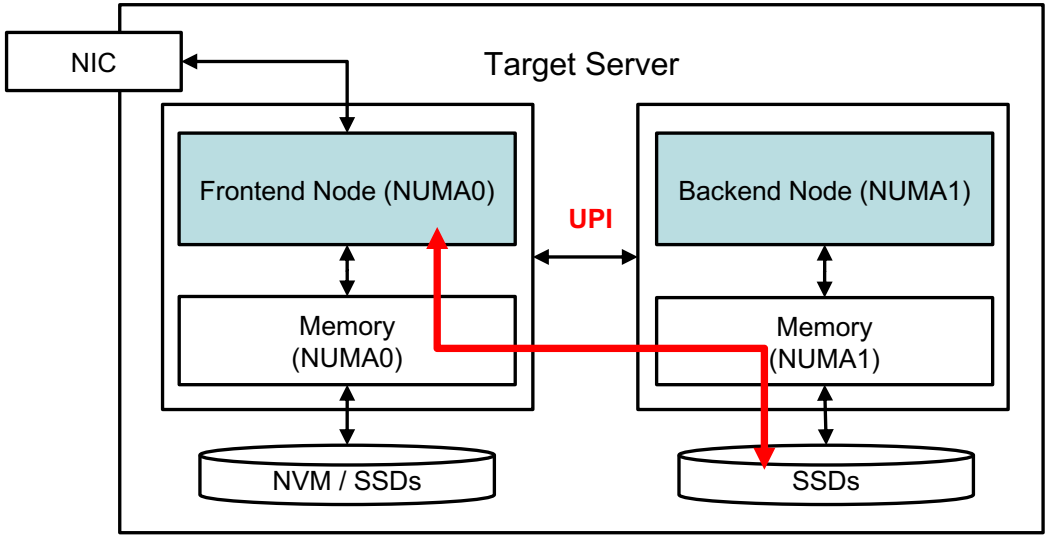
- **Approach 1: Separates CPU sockets for front-end and back-end**
  - Minimizes UPI transactions to spend more CPU cycles for storage services



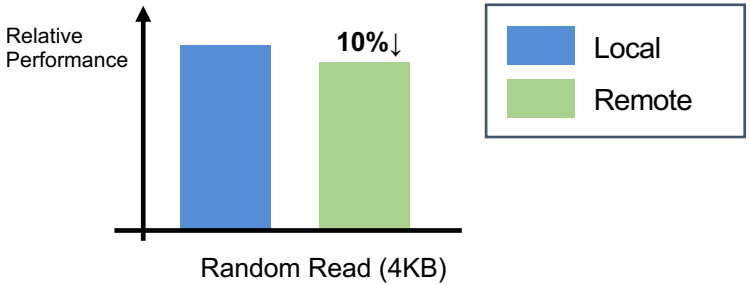
- NUMA0: User IO, Network,
- NUMA1: Flush, RAID, GC, …

Challenge 1: Saturate high-bandwidth network in TCP

- **Approach 1: Separates CPU sockets for front-end and back-end**
  - Minimizes UPI transactions to spend more CPU cycles for storage services

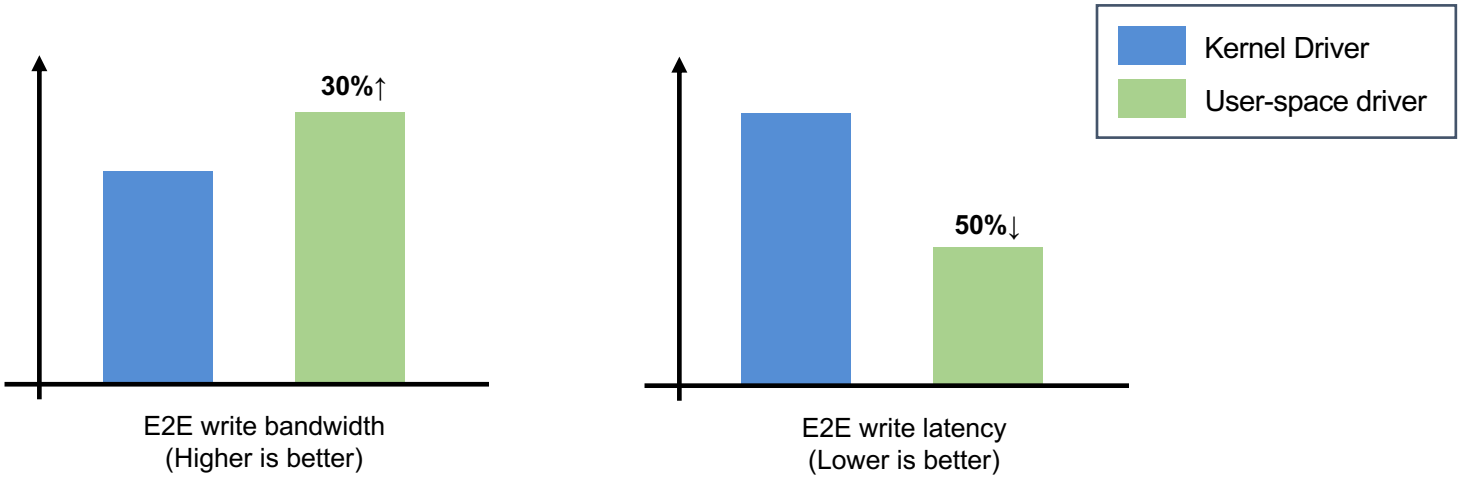


- **Remote SSD access also should be minimized**



Challenge 2: Initiator SW Stack

- **Initiator SW stack does matter to exploit NVMe-oF performance**
  - In case of reads, both kernel and user-space drivers can be achieved max performance
  - In case of writes, only user-space drivers can meet max performance



Other Challenges?

- **Future Works**
  - More NUMA-aware architecture
  - Performance measurement on PCIe Gen4 server
  - More storage feature supported
  - E2E SSD Optimization

- **Will be open-sourced @end of this year!**

Thank You