SK Telecom: A Shareable DAS Pool using a Low Latency NVMe Array

Eric Chang / Program Manager / SK Telecom
Before We Begin...

- SKT NV-Array (NVMe JBOF) has been evolving..

**OCP US Summit 17**

**OCP US Summit 18**

**D20: 1U20**

**E24: 2U24**
NV-Array Demands and Basic Architecture
Increasing Demands for Efficient Infrastructure

- Advanced applications, with significant resource requirements, are becoming ready for deployment:
  - UHD video streaming requires double the bandwidth of full HD ($20\text{Mbps} \times 20\text{K users} = 400\text{Gbps}$)
  - Virtual/augmented Reality based services will evolve to beyond 4K (i.e. 8K to 12K) 360-degree res.
  - 5G wireless communications needs 1/10 latency compared to 4G LTE

- Composable infrastructures are emerging in order to maximize the utilization of these resources:
  - Dynamic reconfiguration of compute, storage and networking allows for the optimal combination of hardware for a specific application

Storage with large capacity, low latency, high bandwidth and composability is a key component of the recently required infrastructure
The NV-Array is designed for high availability, with redundant PCIe switch boards:
- 24 dual port NVMe SSD slots
- Base Management Controller with Redfish and IPMI
- 10 Upstream (Host) Ports

The Host Bus Adaptor provides PCIe cable connectivity to the NV-Array (on COTS servers):
- PCIe x8 and x16 host slot options
- A single HBA can provide two cables to the NV-Array for HA support
SKT's software stack allows data stored in the NV-Array to be shared among multiple host servers.
Key Features and Progress
Data Sharing – SKT Driver and GFS2 (Distributed FS)

- SKT software makes the NV-Array into a shareable DAS pool by:
  - Enabling data sharing among hosts connected to the NV-Array (NTB and GFS2)
  - Managing failover and hardware resources by health monitoring
  - Enhancing storage performance by distributing data traffic between 10 host connections
SKT NV-Array Device Driver (NDD)

- The NDD is a key enabler for SKT’s NVMe based shareable storage system
  - It enables the connection of multiple NVMe SSDs to multiple host servers using the Non Transparent Bridge functions of the PCIe fabric

Reliability - PCIe Hot-Plugging

- The ability to reliably add and remove NVMe SSDs is essential for high availability systems
  - In PCIe terminology, these SSDs must be “hot-pluggable” and the overall system must support “hot-plug”

- The reliable operation of hot-plug work relies on the coordinated interaction between a number of system elements:
  - The system BIOS must support correct system resource allocation for the SSDs, before and after a hot-plug event
  - The Linux kernel must include the proper drivers to support hot-plug, and PCIe error containment and recovery (especially Downstream Port Containment - DPC)
  - The kernel must be correctly configured to allow the BIOS and drivers to work together properly

PCIe Hot-plugging creates dependencies between hardware, BIOS, and kernel versions
Performance - NVMe Multi-path Active/Active Implementation

- SKT improves NVMe multi-path productivity by enabling round-robin path selection
- Dual port NVMe SSD are used in active-active mode rather than active-standby, significantly improving performance

**Namespace structure**

- /dev/nvme0n1
  - (head) namespace
  - `nvme0n1`
  - `nvme0c0n1`
  - `nvme0c1n1`

- Hidden namespace or path

**Conventional vs. Improved implementation**

**Conventional:** One path out of two used

- Commands and data use a single path, until it fails

- RCU List

- If `nvme0c0n1` is NULL or not in NVME_CTRL_LIVE state, the path is routed to `nvme0c1n1`

- In a normal case, only one path is used.

- Underutilized!

**SKT Improvement:** Both paths are used

- Commands and data are distributed between paths until a path fails

- RCU List

- Both paths are used as active/active

- Fully utilized!

Dual port NVMe SSD
Performance Comparison

- SKT’s active/active implementation has made apparent significant performance variations between SSDs
  - Some vendor’s SSDs are not optimally designed for active-active use

By enabling active-active, Read and Write performance improves significantly

![Graph showing multi-path improvement between Default and SKT enabled driver]
SKT has repaired a problem in the current NVME Linux multipath driver:

- When multipathing is enabled, each NVMe subsystem creates a head namespace (e.g., nvme0n1) and multiple hidden namespaces (e.g., nvme0c0n1 and nvme0c1n1) in sysfs.
- When links for hidden namespaces are created while head namespace are used, the namespace creation order must be followed as head namespace and hidden namespace (e.g. nvme0n1 -> nvme0c1n1)
- If the order is not kept, links of sysfs will be incomplete or kernel panic will occur.

Commit link: https://github.com/torvalds/linux/commit/9bd82b1a4418d9b7db000bf557ed608f2872b7c9
Composability - Redfish

- To maximize datacenter efficiency, there is a need to dynamically join disaggregated hardware into complete systems
  - This “composed” system contains the optimal compute, memory, I/O and storage capabilities for a particular workload.
  - Resources can be added and removed without physical interaction with the hardware

- Redfish Composability provides a standard method to manage composed systems

- The Redfish specifications provide data models for composable hardware, and define an interface to manage their composition/decomposition

- A client communicates with a Redfish server using a RESTful interface over HTTPS
  - Data is in JSON format based on OData v4

- Based upon the client’s request, the server will alter the hardware’s state (routing paths, stored parameters, etc.) to adjust the composition

SKT NV-Array supports Redfish for NVMe storage composability

Note) SKT’s other EW session talks about the composability and manageability of system resources in Telco infrastructure
  - Hardware Monitoring and Management System for Telco Data Center (Jungsoo Kim)
Target Apps and Test Results
Target Applications

- High res (i.e. 4K UHD) media streaming / video editing
  - UHD media editing requires 4x the I/O resources of FHD
  - Using the NV-Array dramatically reduces this time consuming process
  - The gains are even larger for Augmented/Virtual Reality infrastructures, with resolutions of 8k or more

- Virtual desktop infrastructure
  - Deduplication for VDI can be achieved by NV-Array using sharing capability

- Real time data analytics
  - Allows in-memory stream processing to be moved to flash, greatly improving capacity

- AI and Deep learning infrastructures
  - Distributed filesystem clusters can be accelerated with the NV-Array

- 5G infrastructures
  - Provides massive, low latency messaging for the network core as well as the billing system
NV-Array based infrastructure system can cover up to hundreds of TB as NVMe SSD capacity scales.

For mid-scale infrastructures, the system with NV-Array will be more cost-effective.
Case 1 - Content Delivery Application

- **Test Environment**
  - 32 client servers (320Gbps load)
  - 8 Host nodes + NV-Array (24 NVMe SSDs)

- **Results**
  - Using the JMeter test tool, the NV-Array system saturated the network bandwidth of **320Gbps**
  - An All-Flash NAS system provided only 50Gbps

---

**Bandwidth**

- **320Gbps**
  - \(10\text{GbE} \times \text{1-port} \times \text{32-server}\)

- **512Gbps**
  - (effective 420Gbps)
  - \(\text{PCIe Gen3 64lanes} = \text{8lanes} \times \text{8-server}\)

**Software setup**

- JMeter Workload Generator (HLS based)
- Nginx + GFS2 + SKT Driver
- 24*NVMe SSD

---

**Diagram Notes**

- 32 client servers
- 10GbE Switch
- COTS Server 8 node
- NV-Array
- 24*NVMe SSD

---

**Calculation**

\[
320 \text{ Gbps} = 10 \times 4 \times 8 = 512 \text{ (effective 420 Gbps)}
\]
Case 2 - VDI Application

- One NV-Array supports up to ten host servers and one thousand VMs (VDI users)
  - Each user is allocated 2K IOPS (3R:7W mix workload)
- The NV-Array IO bandwidth is so high that that user productivity is constrained by CPU performance
  - Service providers can select the appropriate CPU depending on the end user requirements

Note: if raw images are used (relieving CPU bottlenecks), it is expected to provide over 1GBbs/server
Future Work

- NV-Array will be more stable and reliable through testing and real deployment in 2018.

- SKT will keep sharing the experience and identified requirement while verifying PCIe hot-plugging, and contribute NVMe Multi-path driver improvement.

- SKT has a plan to share NV-Array spec and design in OCP around Q4’18.
  - SKT has shared the ‘AF-Media’ hardware design in 2016 and we now offer NV-Array to provide the next-level performance and efficiency by coupling with COTS servers for applications that used ‘AF-Media’.
Summary
Key Takeaway

- There are significant challenges in supporting emerging applications such as 4K UHD, VR, VxI (VDI/VSI/VMI) and 5G infrastructures. Conventional systems, and especially storage, must change to meet these challenges.

- Not only effective capacity and reliability, but low latency and composability are key factors for next generation storage systems.

- All-Flash storage is being re-defined around the advantages of NVMe SSDs. SKT’s NV-Array can usher in a new era of all-Flash storage for the data center.
Other SKT Sessions

- **Hardware Monitoring and Management System for Telco Data Center (Jungsoo Kim)**
  - Date/Time: Wednesday March 21, 9:30am - 10:00am
  - Room: 210 G
  - Engineering workshop: Telco