An Approach for Implementing NVMeOF based Solutions

Sanjeev Kumar
Software Product Engineering, HiTech, Tata Consultancy Services

25 May 2018
Agenda

- Recap of NVMeOF Evolution
- Network Fabric – Which one to Choose?
- Implementing NVMeOF based solution
- Drivers support for NVMeOF
Recap of NVMeOF Evolution
Recap of Communication Protocols

- SAS and SATA performance increased over time but protocols have not changed much.
- NVMe was created to allow direct access to the logical block based architecture of SSDs and to do highly-parallel IO enabling SSD to execute more IO threads than any other protocol before.
Data Flow with Enterprise Storage Over Network: Limitations

Protocol conversion bridge is required to access the data over network which increases the NVMe latency.
What is a NVM Subsystem?

Diagram:
- Network Fabric
- Fabric Port
- NVMe Controller
- NVMe Namespace
- NVMe I/F
- NVM Media
NVMe Over Fabric Solution

- Launched a new specification “NVMe Over Fabric 1.0” on June 2016.
- Is a way to send NVMe commands over networking protocols (“Fabrics”). E.g.
  - RDMA (Infiniband, iWarp, RoCE, ..)
  - Fibre Channel
- Defines a common architecture that supports a range of storage networking fabrics for NVMe block storage protocol over a storage networking fabric
- Inherent parallelism of NVMe multiple I/O Queues is exposed to the host (64k Queues & 64k commands per Q)
- NVMe commands and structures are transferred end-to-end
- Maintains the NVMe architecture across a range of fabric types
- Maintains architecture and software consistency between fabric types by standardizing a common abstraction and encapsulation definition

Design goal of NVMe over Fabrics:
Provide distance connectivity to NVMe devices with no more than 10 microseconds (μs) of additional latency
Network Fabric – Which one to Choose ?
NVMe Over fabric Protocol – Transport Mode

Source: nvmexpress.org
NVMe Over Fabric Stack Architecture

- **NVMe Over Fabric**
  - NVMe Architecture, Queuing Interface, Admin Command & I/O Command Sets, Properties

- **Transport Binding Specifications**
  - Fabric Specific Properties
  - Transport Specific Features/Specialization

- **NVMe Transport**
  - NVMe Transport Binding Services

- **Fabric**
  - NVMe Transport
  - Fabric Protocol

- **Fabric Physical**
  - (i.e. Ethernet, Infiniband, Fiber Channel)
Network Fabrics

- NVMe Over Fabric describe how to transport the NVMe interface across several scalable fabrics
- NVMe Over Fabric initially defines two type of fabrics for NVMe transport as Fiber Channel and RDMA
## Comparative Analysis

<table>
<thead>
<tr>
<th>RoCEv2</th>
<th>iWARP</th>
<th>Infiniband</th>
<th>FCoE</th>
<th>FC</th>
<th>iNVMe</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDMA over Converged Ethernet v2</td>
<td>Internet Wide Area RDMA Protocol</td>
<td>Infiniband</td>
<td>Fiber Channel over Ethernet</td>
<td>Fiber Channel</td>
<td>NVMe Over TCP</td>
</tr>
<tr>
<td>Promoted by Mellanox</td>
<td>Promoted by Intel</td>
<td>Promoted by Mellanox</td>
<td>Promoted by CISCO</td>
<td>Promoted by CISCO, Brocade</td>
<td>Promoted by Facebook, DELL EMC and Intel</td>
</tr>
<tr>
<td>RDMA based</td>
<td>RDMA based</td>
<td>RDMA based</td>
<td>Leverage FC- NVMe</td>
<td>Non RDMA based</td>
<td>Non RDMA based</td>
</tr>
<tr>
<td>Not Compatible with Other Ethernet Options</td>
<td>Not Compatible with Other Ethernet Options</td>
<td>Very Rarely Deployed. Special usecases like HPC or Server-Server cluster</td>
<td>Not Compatible with Other Ethernet Options</td>
<td>Compatible with both SCSI and NVMe protocol</td>
<td>Not Yet a part of NVMe of standard</td>
</tr>
<tr>
<td>Lossless Ethernet support (ECN-Explicit Congestion Notification)</td>
<td>Lossless Ethernet not required</td>
<td>Already a lossless protocol</td>
<td>Require a DCB network (Lossless Ethernet)</td>
<td>Already a lossless protocol</td>
<td>Leverage s/w implementation of NVMe</td>
</tr>
</tbody>
</table>
- Three dominant protocol
- Fiber Channel is a lossless network fabric, NVMe is just a new upper layer protocol where RDMA is not needed
- RDMA + IP + Lossless Ethernet (DCB) layer add complexity to RoCEv2 & iWARP protocols
- Ethernet /IP based protocol are using commodity and internet scale.
Business Drivers to Adapt NVMeOF based Solution

- AI/ Machine Learning
- Analytics
- Video Processing
- High Performance Computing
- Hyper Converged Infrastructure
Implementing NVMeOF based Solution
Core Elements for Implementing NVMe Over Fabric Solutions

1. NVMe Enabled Host System
   - Windows
   - Linux
   - Unix
   - Solaris
   - VMware

2. NVMe Supported Network
   - Fiber Channel
   - RoCE v2
   - iWARP

3. NVMe based Storage Sub System
   - NVMe Controller
   - NSID (NVMe Name Space ID)
   - NVM Interface
   - MVM Media (Chip, SSD, NVDIMM)
Reference Architecture for NVMeOF Solution for Enterprise Storage

- **Linux Host**
  - iSCSI
  - SMB
  - FC

- **Windows Host**
  - iSCSI
  - iSER
  - SRP
  - FC

- **NVMe Optimized Host**
  - NVMe

- **Front End Fabric**
  - NVMe Over Fabric

- **Flash Storage**
  - SCSI → NVMe Bridge
  - NVMe Bridge → NVMe SSD

- **Back End Fabric**
  - NVMe Subsystem
  - NVMe Bridge
  - NVMe SSD
  - PCIe

- **NVMe Subsystem**
  - NVMe Bridge
  - NVMe SSD
  - PCIe
Linux Based NVMeOF Driver
NVMe Over Fabric Support in Linux

- Current functionality implemented for NVMe Host Driver
  - Support for RDMA transport (Infiniband/ RoCE/iWARP)
  - Connect/Disconnect to multiple controllers
  - Transport of NVMe commands/data generated by NVMe core
  - Initial Discovery service implementation
  - Multi-Path

- Current functionality implemented for NVMe Target Driver
  - Support for mandatory NVMe core and Fabrics commands
  - Support for multiple hosts/subsystems/controls/namespaces
  - Namespaces backed by Linux block devices
  - Initial Discovery service; Discovery Subsystem/Controller(s)
  - Target Configuration interface using Linux configfs
  - Create NVM and Discovery Subsystems

Linux Fabrics Driver is a part of Linux Kernel 4.8 onwards
New Fabric Driver uses the existing common code
- Additional it is split into a small common fabrics library and the actual transport driver
- Existing user space APIs of the PCIe driver are also supported when using fabrics
- Uses new sub-command of the existing nvme-cli tool to connect to remote controllers
NVMe Over Fabric Target Driver Implementation in Linux

RDMA Transport

- NVMe Over Fabric Discovery Service
  - Responsible for discovering NVMe Subsystems available to Host, Namespace and multipath

- NVMe Over Fabric Support
  - Responsible for servicing Fabrics commands (connect, property get/set)

- NVMe I/O Command Implementation

- NVMe Admin Command Implementation
  - Admin: Responsible for parsing and servicing admin commands such as controller identify, set features, keep-alive, log page, ...
  - I/O: Responsible for performing the actual I/O (Read, Write, Flush, Deallocate).

- NVMe Target Core
  - Defines and manages the NVMe entities (subsystems, controllers, namespaces, ...) and their allocation
  - Responsible for initial commands processing and correct orchestration of the stack setup and tear down

Linux RDMA Drivers

- iWARP
- RoCE v2
- Infiniband
How Data Flows from Host to Target in NVMeOF?

Host
- Linux File System
- Linux Block IO
- NVMe Device Driver
- NVMe Fabrics
- RDMA Verbs
- RDMA Driver
- RDMA Offload
- TCP Offload

Target
- NVMe Fabrics
- RDMA Verbs
- RDMA Driver
- NVMe Device Driver
- NVMe Drive
- NVMe Drive

Kernel

NIC
NVMe Supported Storage Array Today

- NetApp – E570 All Flash Array (FC-NVMe)
- PureStorage - DirectFlash
- DELL EMC
- IBM
- Supermicro
- Mangstor
- E8 Storage
- Pavillion Data
- Excelero
- Aperion
Q&A
mail us @ sanjeev24.k@tcs.com