NVMe Over Fabric and Direct Attached Host with NVMe SSD SR-IOV

Brian Pan
Aug 2018
Complexity and Bottleneck

Power sequence, bus number, memory address, hotplug, OS and driver

Performance bottleneck
System Architecture-- Direct PCIe and NVMe-oF
System Specification

• Host connection
  – 2x 100G Smart NIC for NVMe over fabric
  – 2x PCIe Gen3 x16 for host connection

• NVMe SSD
  – 16x U.2 NVMe SSD (PCIe Gen3 x4)

• System specification
  – 2U form factor
  – 1+1 redundant 1400W
  – 4+1 redundant fan
### NVMe SSD Pooling by PCIe Switch

#### Spec Details

<table>
<thead>
<tr>
<th>Spec</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>2U</td>
</tr>
<tr>
<td>Front Port</td>
<td>16 U.2 NVMe SSD</td>
</tr>
<tr>
<td>Back Port</td>
<td>Stingray, management, power</td>
</tr>
<tr>
<td>Power</td>
<td>1400W, 1+1 Redundant</td>
</tr>
</tbody>
</table>

- **SMART NIC– Stingray**
  - ARM SoC with 100G smart NIC for ROCEv2

- **NVMe SSD device allocation**
  - Dynamically assign VF of NVMe SSD to smart NIC

- **Hot-plug NVMe SSD**
  - Remove/ add/ re-allocate VF of NVMe SSD from one NIC to another NIC without system shutdown

- **Host-device port configuration**
  - Assign the PCIe slot as PCIe host connection or device ports dynamically (restart required)

- **Management API**
  - Follow the redfish standard and integrate with Intel RSD management software
System
- 1x 25G Smart NIC
- 16x NVMe SSD with SR-IOV capability
- 1x 96 Lane PCIe switch
- PF of all NVMe SSD are installed in PCIe mCPU
- Each PF is with 5 VF. 5 VF are assigned to Smart NIC and host.
Demo of NVMe SR-IOV

1. Create Namespace in NVMe
2. Map NS with VF
3. Assign VF to Smart NIC/ host
4. Run FIO test

Allocate another VF to port14. Port14 is NVMe-over Fabric by Smart NIC.
## FIO Test Result of Direct Attach vs Smart NIC

<table>
<thead>
<tr>
<th>Performance</th>
<th>Throughput</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directly attach to host</td>
<td>2853.5 MB/s</td>
<td>345.03 usec</td>
</tr>
<tr>
<td>NVMe over Fabric</td>
<td>2588.9 MB/s</td>
<td>380.49 usec</td>
</tr>
</tbody>
</table>
NVMe with or without SR-IOV Capability

NVMe with SR-IOV. VM talk to VF directly. PF is sit on PCIe switch.

NVMe without SR-IOV. Passthrough model

NVMe without SR-IOV. Hypervisor manage VMs to NVMe SSD
## Performance Results of NVMe with SR-IOV

**NVMe Virtual Functions on Linux KVM (Passthrough)**

### Read Performance

<table>
<thead>
<tr>
<th>Tested Functions</th>
<th>4K Read (Random)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MB/s</td>
</tr>
<tr>
<td>VM_1 access to VF_1</td>
<td>280</td>
</tr>
<tr>
<td>VM_2 access to VF_2</td>
<td>280</td>
</tr>
<tr>
<td>VM_3 access to VF_3</td>
<td>277</td>
</tr>
<tr>
<td>VM_4 access to VF_4</td>
<td>283</td>
</tr>
<tr>
<td>VM_5 access to VF_5</td>
<td>281</td>
</tr>
<tr>
<td>VM_1 access to VF_1</td>
<td>310</td>
</tr>
</tbody>
</table>

The performance measured using `fio` in CentOS 7.5, with queue depth 16 by 1 worker.
# Performance Results of NVMe without SR-IOV

**NVMe Physical Function on Linux KVM (Shareable)**

<table>
<thead>
<tr>
<th>Tested Functions</th>
<th>4K Read (Random)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MB/s</td>
</tr>
<tr>
<td>VM_1 access to PF</td>
<td>90</td>
</tr>
<tr>
<td>VM_2 access to PF</td>
<td>97</td>
</tr>
<tr>
<td>VM_3 access to PF</td>
<td>92</td>
</tr>
<tr>
<td>VM_4 access to PF</td>
<td>95</td>
</tr>
<tr>
<td>VM_5 access to PF</td>
<td>90</td>
</tr>
<tr>
<td>VM_1 access to PF</td>
<td>119</td>
</tr>
</tbody>
</table>

The performance measured using Fio in CentOS 7.5, with queue depth 16 by 1 worker.
Key Benefits of NVMe SR-IOV

• Performance
  – The VF latency is only ⅓ of PF latency in multi-VMs environment
  – The performance is 3 times of PF performance
  – PCIe Gen3 x4 performance can be shared by multi-VMs

• Cost saving
  – Tens of VFs associated with a single PF, extending the capacity of a device and lowering hardware cost
  – With better latency and performance, the utilization rate will be higher to further reduce the hardware cost
  – Reduce NVMe SSD amount by sharing NVMe via PCIe
Key Benefits of NVMe SR-IOV

- **Multi-path IO via PCIe**
  - The name space on NVMe can be accessed by different hosts through PCIe connection

- **Flexibility configuration**
  - Dynamic control by the PF through registers designed to turn on the SR-IOV capability, eliminating the need via direct access to hardware from the virtual machine environment.

- **Inter-operatability**
  - A standard way of sharing the capacity of any given I/O device thus allowing for the most efficient use of that resource in a virtual system
Brian Pan | H3

GM

huaiyangpan

www.h3platform.com

brian.pan@h3platform.com

+886 2 2698 3800#110