100G NVMe over Fabrics for AMD EPYC

Using AMD EPYC 7551 Platform & Chelsio T6 iWARP RDMA Adapter

Executive Summary

NVMe over Fabrics specification extends the benefits of NVMe to large fabrics, beyond the reach and scalability of PCIe. NVMe enables deployments of hundreds or thousands of SSDs using a network interconnect, such as RDMA over Ethernet. T6 iWARP RDMA provides a low latency, high throughput, plug-and-play Ethernet solution for connecting high performance NVMe SSDs over a scalable, congestion controlled and traffic managed fabric, with no special configuration needed.

AMD EPYC, industry's first hardware-embedded x86 server security solution, is a system on chip (SoC) which provides exceptional processing power coupled with high-end memory and I/O resources to meet workload demands of any scale, from virtualized infrastructures to cloud-era datacenters. The combination of the AMD EPYC 7551 server with Chelsio's industry-leading Unified Wire adapter solution delivers compelling performance, power and total cost of ownership (TCO) advantages. This enables innovative topologies and networked computing models to address the most demanding processing needs. This paper presents the performance results of Chelsio NVMe-oF over 100GbE iWARP fabric in an AMD EPYC 7551 Server (an x86 platform) setup. The Chelsio NVMe solution delivers line-rate throughput of 99 Gbps and more than 2.4M IOPS.

Test Results

The following graph presents READ, WRITE IOPS and throughput performance of Chelsio NVMe-oF solution using null block device as storage array. The results are collected using the **fio** tool with I/O size varying from 4 to 512 Kbytes with an access pattern of random READs and WRITEs.

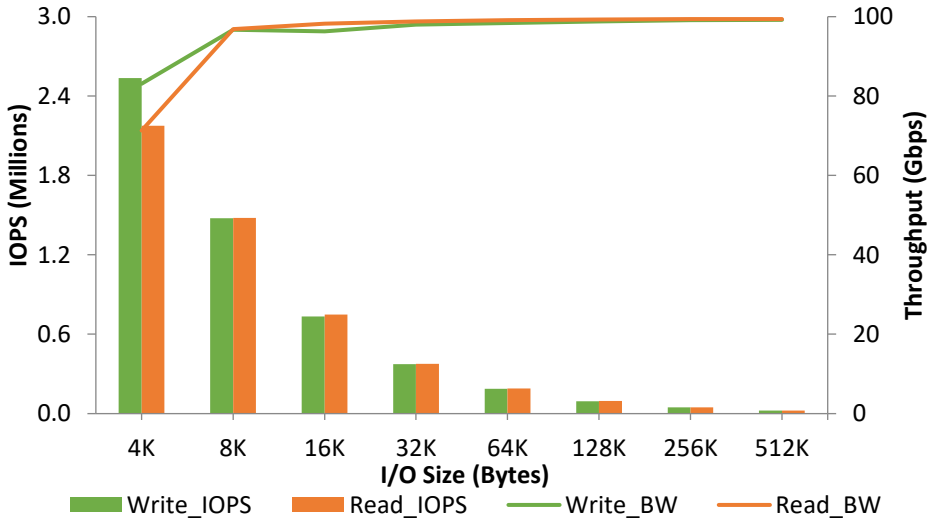


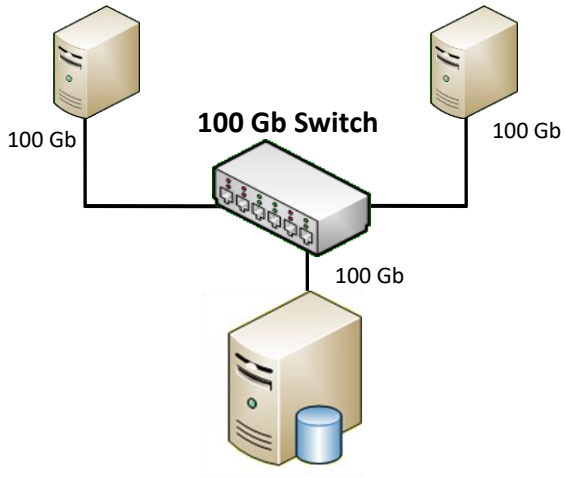
Figure 1 – READ, WRITE Throughput & IOPS vs. I/O size

T6 NVMe solution delivers a staggering 99 Gbps line-rate throughput performance for both READ and WRITE. In addition, WRITE IOPS performance exceeds 2.4M. The results above were achieved with the target CPU utilization of ~5% for 512 Kbytes I/O size and up to ~20% for 4 Kbytes I/O size.

This highlights the value of NVMe-oF offload, where high IOPs and high throughput can be achieved with low CPU utilization. This in turn frees up the server node for application use.

The Demonstration

The setup consists of an NVMe target machine connected to 2 initiator machines through a 100GbE switch using single port on each system. MTU of 9000B was used. Latest Chelsio Unified Wire driver was installed on each machine.



- Supermicro X10DRG-Q Initiators with T62100-CR adapter
- 2 Intel Xeon CPU E5-2687W 0 12-core @ 3.00GHz (HT disabled)
- 128GB RAM
- RHEL7.4 (4.14.67 kernel)

- Supermicro H11DSi-NT Target with T62100-CR adapter
- 1 AMD EPYC 7551 32-core processor (HT disabled)
- 128GB RAM
- RHEL7.5 (4.14.67 kernel)

Figure 2 – Test Setup

Storage configuration

The target is configured with 4 null block devices, each of 1GB size. Each initiator uses 2 connections.

Setup Configuration

Following performance tunings were done on AMD EPYC server:

i. Updated BIOS to latest version (v1.1b was used in this case).
ii. BIOS Settings:
SVM (Virtualization), Global C-state Control, Hyperthreading, IOMMU, SR-IOV and Core Performance Boost were *Disabled.*
Determinism Slider was set to *Performance*
Memory Interleaving was set to *Auto*
iii. All the memory channels were populated with maximum supported speed.
iv. Added *'iommu=pt cpuidle.off=1 processor.max_cstate=0'* to the kernel command line to disable c-states and rebooted the machine.

Target/Initiator Configuration

Following configuration was done on all the machines:

i. Compiled and installed 4.14.51 kernel from https://github.com/larrystevenwise/linux, linux-4.14-nvme branch.
ii. Added *'processor.max_cstate=1 intel_pstate=disable intel_idle.max_cstate=0'* to the kernel command line on initiator machines and rebooted them.

iii. Installed Chelsio Unified Wire v3.9.0.0.

```
[root@host~]# make CONF=NVME_PERFORMANCE install
```

iv. Following power saving profile was set.

```
[root@host~]# tuned-adm profile network-throughput
```

v. Loaded the Chelsio iWARP RDMA driver.

```
[root@host~]# modprobe iw_cxgb4
```

vi. Chelsio interface was assigned with IPv4 address, MTU 9000 and brought-up.

```
[root@host~]# ifconfig ethX <IP address> mtu 9000 up
```

vii. Loaded the NVMe drivers.

```
[root@host~]# modprobe nvmet
[root@host~]# modprobe nvme
[root@host~]# modprobe nvmet-rdma
```

viii. CPU affinity was set for BW/IOPs test.

```
[root@host~]# t4_perftune.sh -n -Q rdma
```

Target Configuration

i. Created 4 Null Block devices, each of 1GB size.

```
[root@host~]# modprobe null_blk nr_devices=4 gb=1 use_per_node_hctx=Y
```

ii. Configured the target using the below script:

```
#!/bin/bash
nvmetcli clear > /dev/null 2>$1
mount -t configfs none /sys/kernel/config > /dev/null 2>$1
rmmod nvmet_rdma nvmet nvme null_blk  > /dev/null 2>$1
sleep 1
IPPORT="4420"         # 4420 is the reserved NVME/Fabrics RDMA port
IPADDR="102.2.2.238"  # the ipaddress of your target rdma interface
NAME="nvme-nullb"
DEV="/dev/nullb"
for i in `seq 0 3`; do
mkdir /sys/kernel/config/nvmet/subsystems/${NAME}${i}
mkdir -p /sys/kernel/config/nvmet/subsystems/${NAME}${i}/namespaces/1
echo -n ${DEV}${i}
>/sys/kernel/config/nvmet/subsystems/${NAME}${i}/namespaces/1/device_path
echo 1 > /sys/kernel/config/nvmet/subsystems/${NAME}${i}/attr_allow_any_host
echo 1 > /sys/kernel/config/nvmet/subsystems/${NAME}${i}/namespaces/1/enable
done
mkdir /sys/kernel/config/nvmet/ports/1
echo 8192 > /sys/kernel/config/nvmet/ports/1/param_inline_data_size
echo "ipv4" > /sys/kernel/config/nvmet/ports/1/addr_adrfam
echo "rdma" > /sys/kernel/config/nvmet/ports/1/addr_trtype
echo $IPPORT > /sys/kernel/config/nvmet/ports/1/addr_trsvcid
```

```
echo $IPADDR > /sys/kernel/config/nvmet/ports/1/addr_traddr
for i in `seq 0 3` ; do
                ln -s /sys/kernel/config/nvmet/subsystems/${NAME}${i}
/sys/kernel/config/nvmet/ports/1/subsystems/${NAME}${i}
done
```

Initiator Configuration

i. Target was discovered:

```
[root@host~]# nvme discover -t rdma -a <target_ip> -s 4420
```

ii. Initiator1 connected to Target.

```
[root@host1~]# nvme connect -t rdma -i 2 -a 102.2.2.238 -n nvme-nullb0 -s 4420
[root@host1~]# nvme connect -t rdma -i 2 -a 102.2.2.238 -n nvme-nullb1 -s 4420
```

iii. Initiator2 connected to Target.

```
[root@host2~]# nvme connect -t rdma -i 2 -a 102.2.2.238 -n nvme-nullb2 -s 4420
[root@host2~]# nvme connect -t rdma -i 2 -a 102.2.2.238 -n nvme-nullb3 -s 4420
```

iv. *fio* tool was run on both initiators.

```
[root@host~]# fio --rw=randwrite/randread --ioengine=libaio --name=random  --
size=400m --invalidate=1 --direct=1 --runtime=30 --time_based --
fsync_on_close=1 --group_reporting --filename= <device list> --iodepth=64 --
numjobs=16 --bs=<value>
```

Conclusion

This paper showcases the performance capabilities of Chelsio T6 100G NVMe-oF solution in AMD EPYC server setup with T62100-CR Unified Wire adapter. Using iWARP RDMA enables the NVMe based storage to be shared, pooled and managed more effectively across a low latency, high performance network. The results show that Chelsio's NVMe-oF solution achieves:

- Line-rate throughput of 99 Gbps for both READ and WRITE
- IOPS exceeding 2.4M
- Minimal CPU Utilization

Related Links

100G Network Performance for AMD EPYC
FreeBSD 100G TOE Performance for AMD EPYC
100G iSCSI Performance for AMD EPYC