Chelsio 100G Performance for AMD EPYC

Using AMD EPYC™ 7551 Platform & Chelsio T6 Adapter

Executive Summary

AMD EPYC, containing the industry's first hardware-embedded x86 server security processor, 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 virtually any scale, from virtualized infrastructures to cloud-era datacenters. The combination of an AMD EPYC™ 7551 powered 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.

Chelsio T6 is an efficient, high-performance and low-latency unified wire adapter with the unique ability to fully offload TCP/IP, iSCSI and iWARP protocols using a single ASIC and firmware, optimized for storage, cloud computing, HPC, virtualization and other datacenter networking applications. Chelsio adapters unburden communication responsibilities and processing overhead from servers and storage systems by enabling a true converged Unified Wire solution resulting in a dramatic increase in application performance with a minimum of CPU cycles.

This paper demonstrates Chelsio 100G solution delivering line-rate performance in an AMD EPYC powered server environment:

- **Network performance:** Delivering line-rate 99 Gbps throughput for both Transmit and Receive directions.
- **iSCSI offload performance:** 100G iSCSI offload solution delivering 98 Gbps line-rate throughput and more than 2.7M IOPS for a cost-effective enterprise-class storage target solution.
- **NVMe over Fabrics:** The Chelsio NVMe-oF over 100GbE iWARP solution delivers line-rate throughput of 99 Gbps and more than 2.4M IOPS.
- **OVS Kernel Datapath Offload:** Chelsio's T6 offload scores a considerable PPS of 47M with less than 1% CPU usage across the board.

Network Performance

Chelsio adapters provide extensive support for stateless offload operation for both IPv4 and IPv6 (IP, TCP and UDP checksum offload, Large Send Offload, Larger Receive Offload, Receive Side Steering/Load Balancing, and flexible line rate Filtering). This demonstrates Chelsio 100G Network adapter T62100-CR delivering line-rate 99 Gbps throughput for both Transmit and Receive directions in an AMD EPYC Server environment.

Test Results

The following graphs plot the Tx and Rx network performance results varying the number of connections. The results are obtained using Iperf tool across I/O sizes ranging from 64 bytes to 512 Kbytes.

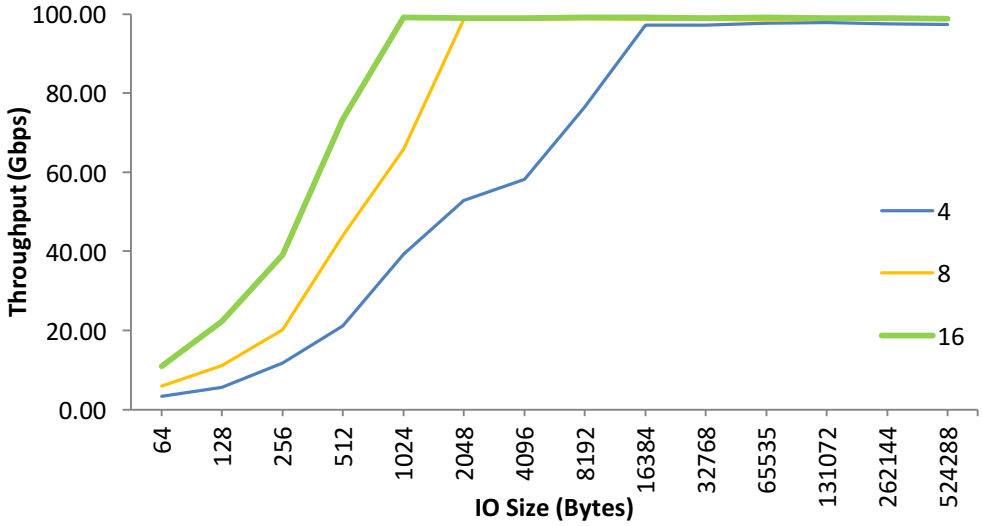


Figure 1 – Tx Performance vs. I/O size

Chelsio adapter delivers line-rate throughput of 99 Gbps even with multiple connections.

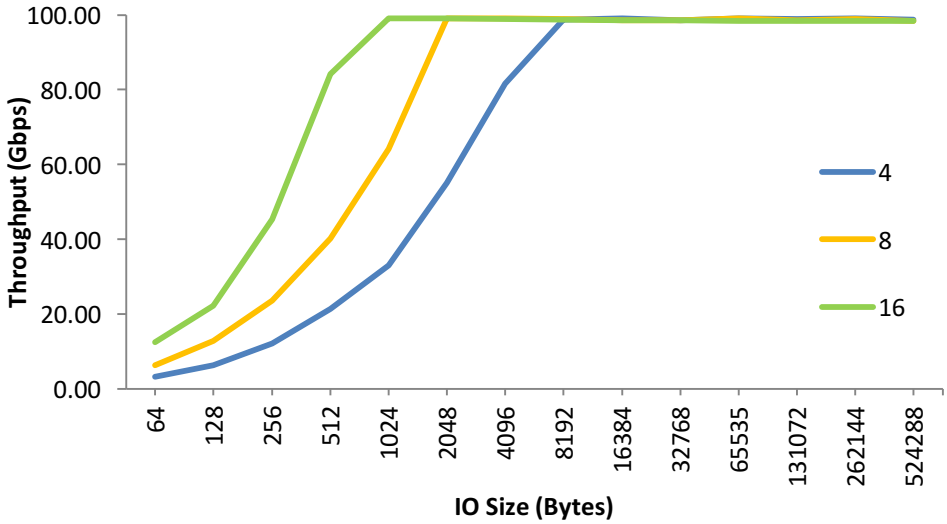


Figure 2 – Rx Performance vs. I/O size

Rx Performance graph reflects similar results, with line-rate throughput of 99 Gbps.

Test Setup

The setup consists of an AMD EPYC powered machine connected to a PEER machine with a single 100G port. MTU of 9000B was used. Latest Chelsio Unified Wire drivers for Linux was installed on both machines.

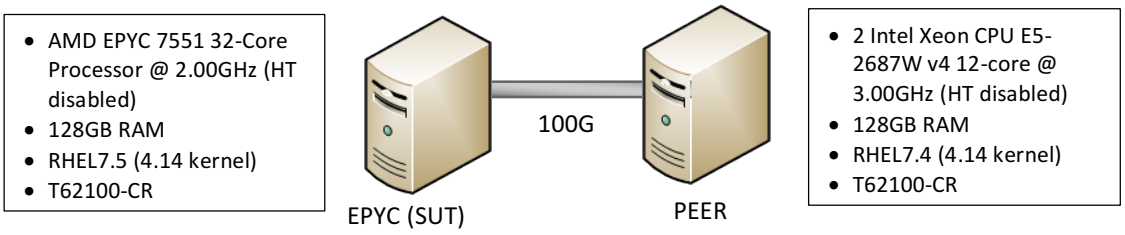


Figure 3 – Test Setup

Setup Configuration

Performance Settings:

Following performance settings were done on
AMD EPYC POWERED SUT (System Under Test):

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.

PEER:

v. BIOS Settings:
Virtualization, Hyperthreading, SR-IOV, IOAT, VT-D and CPU-Power technology were Disabled.

vi. Added 'processor.max_cstate=1 intel_pstate=disable intel_idle.max_cstate=0' to the kernel command line and rebooted.

Other common settings were done on both SUT and PEER:

vii. Following services were stopped.

```
[root@host~]# systemctl stop firewalld.service
[root@host~]# systemctl stop irqbalance.service
```

viii. Following power saving profiles were set.

```
[root@host~]# tuned-adm profile network-throughput
[root@host~]# cpupower frequency-set --governor performance
```

Following configuration was done on all the machines:

i. Installed Chelsio Unified Wire v3.9.0.0.

```
[root@host~]# make install
```

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

```
[root@host~]# modprobe cxgb4
[root@host~]# ifconfig ethX <IP address> mtu 9000 up
```

iii. Enabled adaptive-rx for Chelsio interface.

```
[root@host~]# ethtool -C ethX adaptive-rx on
```

iv. Mapped the Chelsio Interface IRQ's to different CPU cores.

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

v. Following Sysctls were set:

```
[root@host~]# sysctl -w net.ipv4.tcp_timestamps=0
[root@host~]# sysctl -w net.core.netdev_max_backlog=250000
[root@host~]# sysctl -w net.core.rmem_max=1048576
[root@host~]# sysctl -w net.core.wmem_max=1048576
[root@host~]# sysctl -w net.core.rmem_default=1048576
[root@host~]# sysctl -w net.core.wmem_default=1048576
[root@host~]# sysctl -w net.ipv4.tcp_rmem="4096      1048576 1048576"
[root@host~]# sysctl -w net.ipv4.tcp_wmem="4096      1048576 1048576"
```

Commands Used

Server: *iperf -s -p <port>*
Client: *iperf -c <Server IP> -p <port> -l <IO Size> -t 30 -P <# Conn>*

iSCSI Offload Performance

This demonstrates Chelsio 100G iSCSI offload solution delivering 98 Gbps line-rate throughput and more than 2.7M IOPS for a cost-effective enterprise-class storage target solution built with volume, off-the-shelf hardware and software components.

Test Results

The following graph presents READ, WRITE IOPS and throughput performance of Chelsio T6 iSCSI solution using null block device as storage array. The results are collected using the **fio** tool with I/O size varying from 4k to 512k bytes with an access pattern of random READs and WRITEs.

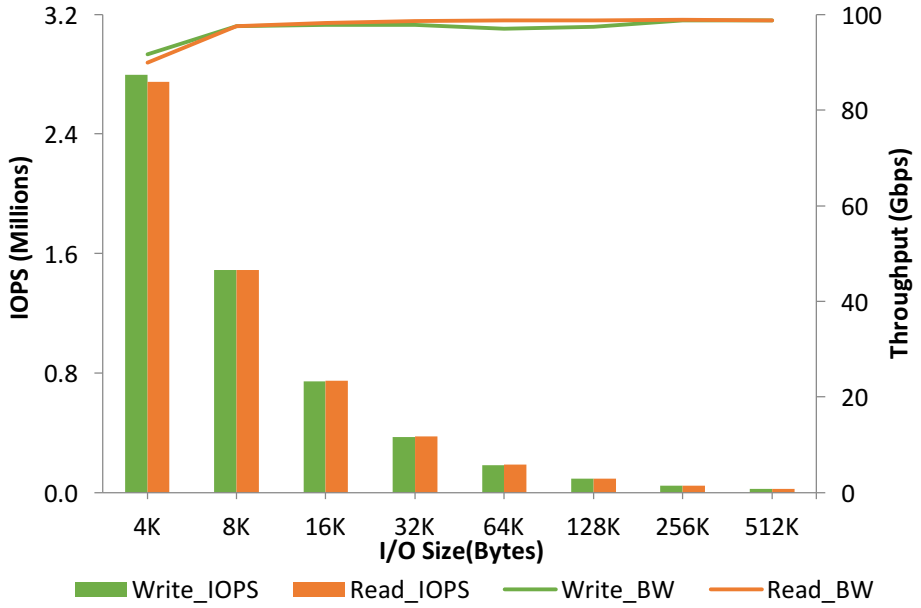


Figure 4 – READ, WRITE Throughput & IOPS vs. I/O size

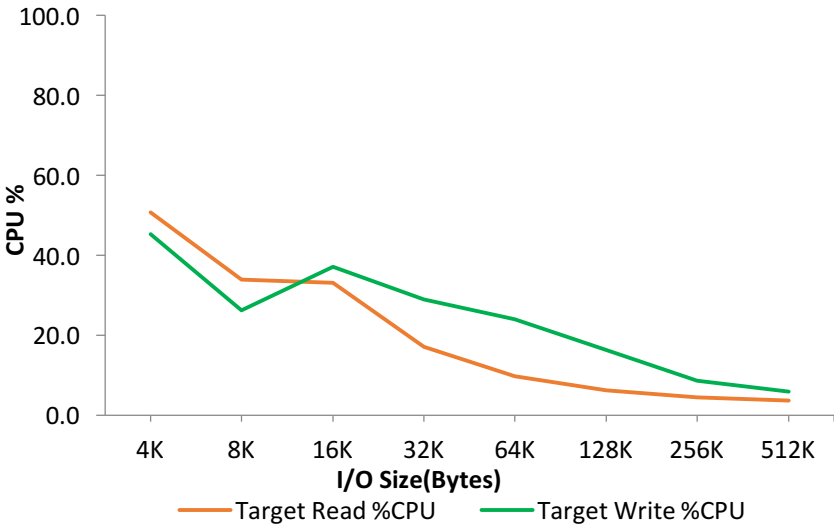


Figure 5 – Target % CPU vs. I/O size

As seen from the above results, T6 iSCSI Offload solution delivers an exceptional line-rate throughput performance of 98 Gbps for both READ and WRITE. CPU savings on the Target is considerable for both READ and WRITE operations, peaking at less than 50% and gradually diminishing as I/O size increases. In addition, the solution scores a commendable IOPS performance of 2.7M. With a consistent and scalable performance, Chelsio iSCSI Offload solution provides an all-round SAN solution for exceptional I/O performance and efficiency.

Test Setup

The setup consists of an LIO iSCSI target machine connected to 4 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.

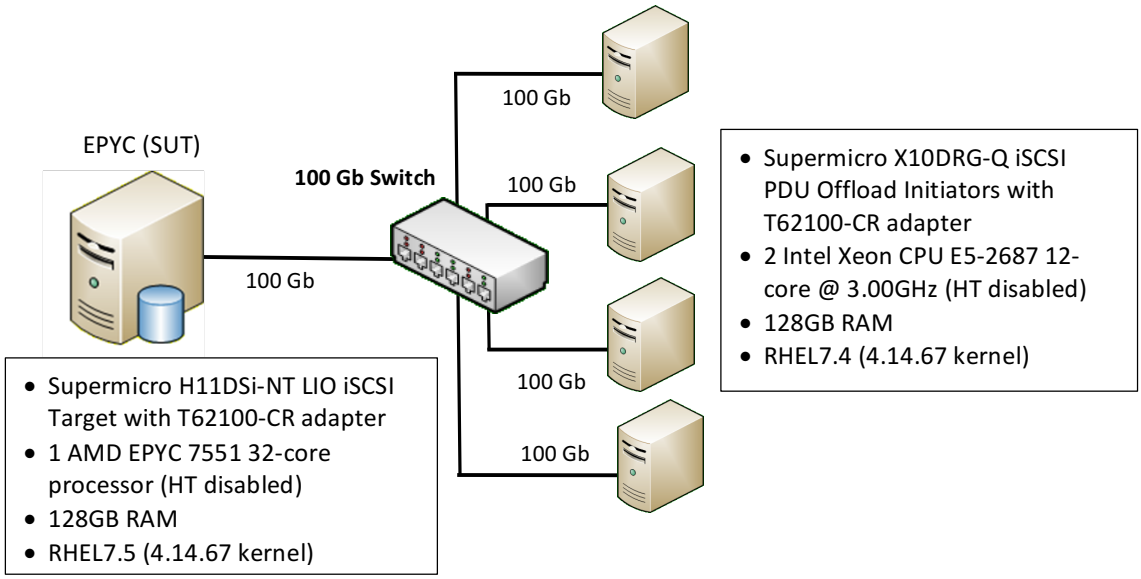


Figure 6 – Test Setup

Storage Configuration

The target was configured in offload mode with 32 Ramdisk (nullio) LUNs, each of 600MB size. Each initiator uses 8 connections.

Setup Configuration

Followed the **Performance Settings** guidelines described in 'Network Performance setup configuration':

Target/Initiator Configuration

Following configuration was done on all the machines:

i. Installed Chelsio Unified Wire v3.9.0.0.

```
[root@host~]# make install
```

Target Configuration

i. LIO iSCSI Offload target driver was loaded.

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

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

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

iii. CPU affinity was set.

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

iv. Configured 32 targets using *targetcli*.

```
[root@host~]# for i in `seq 1 32`; do targetcli /backstores/ramdisk/ create
nullio=true size=600M name=ramdisk$i ; done
[root@host~]# for i in `seq 1 32`; do targetcli /iscsi create iqn.2017-
09.org.linux-iscsi.target$i ; done
[root@host~]# for i in `seq 1 32`; do targetcli /iscsi/iqn.2017-
09.org.linux-iscsi.target$i/tpg1/ set attribute authentication=0
demo_mode_write_protect=0 generate_node_acls=1 cache_dynamic_acls=1  ; done
[root@host~]# for i in `seq 1 32`; do targetcli /iscsi/iqn.2017-
09.org.linux-iscsi.target$i/tpg1/luns create lun=0
storage_object=/backstores/ramdisk/ramdisk$i  ; done
[root@host~]# for i in `seq 1 32`; do targetcli /iscsi/iqn.2017-
09.org.linux-iscsi.target$i/tpg1/portals/ delete ip_address=0.0.0.0
ip_port=3260 ; done
[root@host~]# for i in `seq 1 32`; do targetcli /iscsi/iqn.2017-
09.org.linux-iscsi.target$i/tpg1/portals create ip_address=102.2.2.238
ip_port=3260 ; done
[root@host~]#for i in `seq 1 32`; do targetcli iscsi/iqn.2017-09.org.linux-
iscsi.target$i/tpg1/ set parameter InitialR2T=No; done
```

v. Enabled LIO Target Offload.

```
[root@host~]# for i in `seq 1 32  `; do targetcli /iscsi/iqn.2017-
09.org.linux-iscsi.target$i/tpg1/portals/10.1.1.8:3260 enable_offload
boolean=True ; done
```

Initiator Configuration

i. Added '*scsi_mod.use_blk_mq=Y processor.max_cstate=1 intel_pstate=disable intel_idle.max_cstate=0*' to the kernel command line and rebooted the machine.

ii. iSCSI Initiator driver was loaded:

```
[root@host~]# modprobe cxgb4i
```

iii. Chelsio interface was assigned with IPv4 address and brought-up.

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

iv. CPU affinity was set:

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

v. Target was discovered using *cxgb4i* iface

```
[root@host~]# iscsiadm -m discovery -t st -p 102.2.2.238:3260 -I
<cxgb4i_iface>
```

vi. Logged in to targets from the 4 initiators.

```
[root@host1~]# for i in `seq 1 8`; do iscsiadm -m node iqn.2017-
09.org.linux-iscsi.target$i -p 102.2.2.238:3260 -I <cxgb4i_iface> -l; done
[root@host2~]# for i in `seq 9 16`;do iscsiadm -m node -T iqn.2017-
09.org.linux-iscsi.target$i -p 102.2.2.238:3260 -I <cxgb4i_iface> -l ;done
```

```
[root@host3~]# for i in `seq 17 24`;do iscsiadm -m node -T iqn.2017-
09.org.linux-iscsi.target$i -p 102.2.2.238:3260 -I <cxgb4i_iface> -l ;done
[root@host4~]# for i in `seq 25 32`;do iscsiadm -m node -T iqn.2017-
09.org.linux-iscsi.target$i -p 102.2.2.238:3260 -I <cxgb4i_iface> -l ;done
```

vii. fio tool was run on all 4 initiators for Throughput and IOPS test.

```
[root@host~]# fio --rw=<randwrite/randwrite> --ioengine=libaio --name=random
--size=400m --invalidate=1 --direct=1 --runtime=30 --time_based --
fsync_on_close=1 --group_reporting --filename=<device list>  --iodepth=32 --
numjobs=8 --bs=<value>
```

NVMe over Fabrics

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.

This presents the performance results of Chelsio NVMe-oF over 100GbE iWARP fabric in an AMD EPYC 7551 Powered 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.

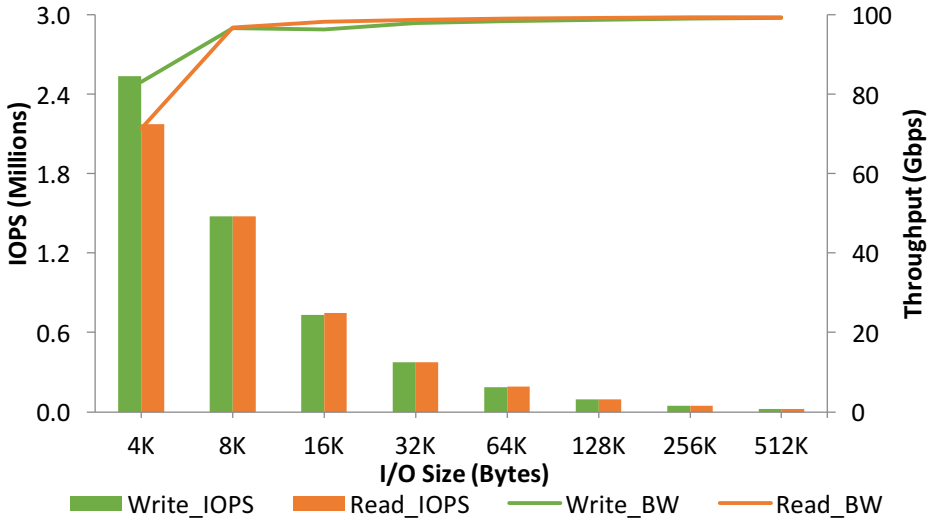


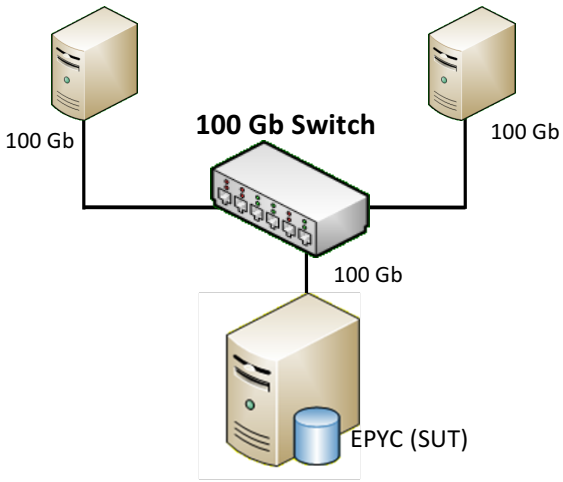
Figure 7 – READ, WRITE Throughput & IOPS vs. I/O size

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.

Test Setup

The setup consists of an NVMe target machine (SUT) 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 8 – Test Setup

Storage configuration

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

Setup Configuration

Followed the **Performance Settings** guidelines described in 'Network Performance setup configuration':

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. Installed Chelsio Unified Wire v3.9.0.0.

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

iii. Loaded the Chelsio iWARP RDMA driver.

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

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

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

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

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

Target Configuration

i. Loaded nvmet and nvmet-rdma kernel modules.

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

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

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

iii. 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. Loaded nvme-rdma kernel module.

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

ii. Target was discovered:

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

iii. 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
```

iv. 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
```

v. *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>
```

OVS Kernel Datapath Offload

This highlights Chelsio's OVS Kernel Datapath offload capabilities in an AMD EPYC 7551 Powered Server (an x86 platform) setup by comparing packet processing rate (PPS) and CPU usage of offloaded and non-offloaded OpenFlow network traffic. Chelsio's T6 offload delivers a considerable PPS of 47M with less than 1% CPU usage across the board.

Test Results

The following graph presents packet processing rate (PPS) and CPU utilization for offload and non-offload OpenFlow network traffic. The results are collected using **pktgen** tool with I/O size 64B and the number of OpenFlows varying from 1 to 10k.

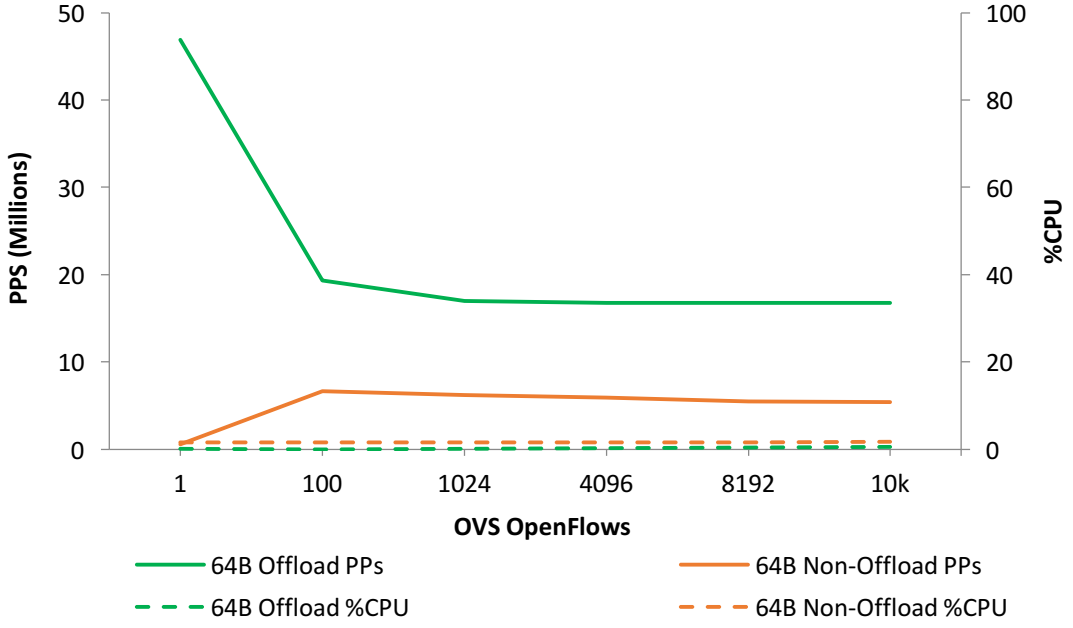


Figure 9 - PPS and %CPU vs. Number of Flows

By offloading OVS kernel datapath on to the adapter, Chelsio T62100-CR adapter performs exceptionally well with up to 47 MPPS at challengingly small I/O size. As the flows increase beyond

100, Chelsio's offload solution continues to deliver with an average of 3x the performance of non-offload. Furthermore, with CPU usage of less than 1%, the processing power of the server is practically unused; free to be utilized for other CPU intensive applications.

Test Setup

The following diagram provides the test setup and configuration details:

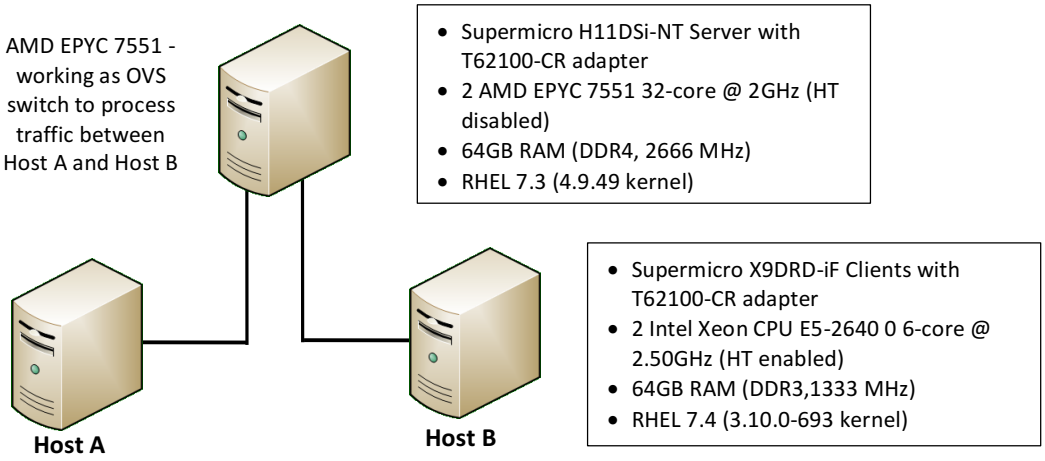


Figure 10 – Test Setup

The test setup consists of 2 Client machines connected to an OVS Switch (AMD EPYC 7551 Powered Server) machine using single 100Gb link. MTU of 9000B is configured on all the machines. Latest Chelsio Unified Wire software is installed on all the machines.

Conclusion

This paper showcases the performance advantages of Chelsio 100G adapter in AMD EPYC 7551 powered server environment for below:

With **network** line-rate throughput of 99 Gbps, Chelsio Unified Wire Ethernet adapters provide the optimal and scalable networking building block for datacenters with AMD EPYC based servers.

Chelsio's **iSCSI offload** software with T6 adapter provides **i**ndustry leading iSCSI performance with the highest IOPS and bandwidth available today. The resulting solution is highly competitive with special purpose systems and storage infrastructure currently on the market in both performance and cost. Chelsio's iSCSI offload solution delivers:

- Line-rate throughput of 98 Gbps for both READ and WRITE
- IOPS of more than 2.7M

Chelsio's T6 100G **NVMe-oF solution 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

Chelsio's T6 **OVS offload solution** delivers up to 3x the packet processing rate of regular (non-offload) NIC with a maximum of 47M. Such an exceptional performance while processing small I/O size (64B) network packets is representative of real-world application demands. In addition to the high packet processing performance, Chelsio's T6 solution delivers extraordinary CPU management capabilities with processing usage never crossing over 0.6%, even with 10k flows.

Chelsio's T6 Unified Wire adapter and AMD's EPYC server platform is a definite answer to the ever-increasing processing demands of datacenters in any environment-bare metal, virtual or cloud.

Related Links

FreeBSD 100G TOE Performance for AMD EPYC
Demartek Evaluation: Chelsio Terminator 6 (T6) Unified Wire Adapter iSCSI Offload