NFS/TCP Offload

Boosting NFS Performance & Efficiency with Chelsio T5/T6 TOE

Executive Summary

Chelsio is the leading provider of network protocol offloading technologies, and its TCP Offload Engine (TOE) is the first and currently only engine capable of full TCP/IP offload up to 100Gbps. Direct Data Placement (DDP) addresses the memory subsystem bottleneck issue on receive, by defining how data can be directly placed into an upper layer protocol's (ULP) receive buffer without intermediate buffers, thus enabling data transfers with minimal CPU utilization. In Transmit, the Chelsio TOE solution supports TCP/IP processing in a cut-through fashion, using Tx Zero-Copy to achieve optimal bandwidth and latency.

NFS is a highly popular method of consolidating file resources in today's complex computing environments using standard TCP/IP networks. NFSv4, the latest version of the NFS protocol, has improvements in security, maintainability, and performance. The Chelsio adapters are designed specifically to perform computationally intensive network operations more efficiently than general-purpose CPUs. Servers with system load comprising of NFS network operations see great savings in CPU Utilization by offloading these operations to the Chelsio adapter.

This paper presents performance comparison of NFS/TCP Server with Chelsio TCP Offload and regular L2 NIC using the Chelsio T62100-CR and T540-LP-CR Unified Wire adapters in FreeBSD. The L2 NIC mode of operation is similar to what is expected from other non-offloaded L2 NIC solutions. The results show:

- Both the adapters deliver line-rate performance in both TCP Offload and L2 NIC modes.
- Remarkable improvements in CPU usage when TCP Offload is in use, compared to L2 NIC.
- Cost savings enabled by TOE DDP solution.

Thanks to an inbox driver in the FreeBSD kernel, T5/T6-based adapters are plug-and-play solutions for extreme networking performance, with concurrent support of feature-rich capabilities like Inline TLS/SSL offload (only T6), Traffic Management, Filtering, iWARP RDMA, SR-IOV Virtual Functions, and iSCSI.

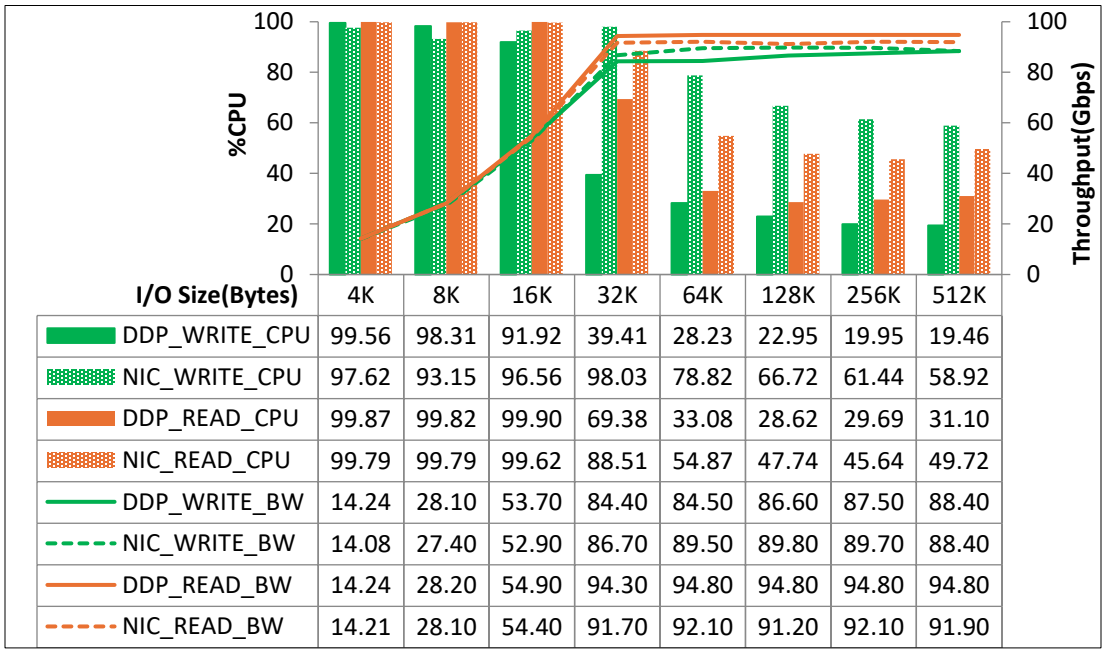
Test Overview

The following tests present the CPU usage and throughput results of NFS/TCP server, using Chelsio TCP Offload and L2 NIC modes. The numbers are collected using *fio* tool with I/O sizes from 4 to 512 Kbytes with an access pattern of Random READs and WRITEs.

1. T62100-CR 1x100G MTU 1500
2. T62100-CR 1x100G MTU 9000
3. T540-LP-CR 4x10G MTU 1500
4. T540-LP-CR 4x10G MTU 9000

Test Results

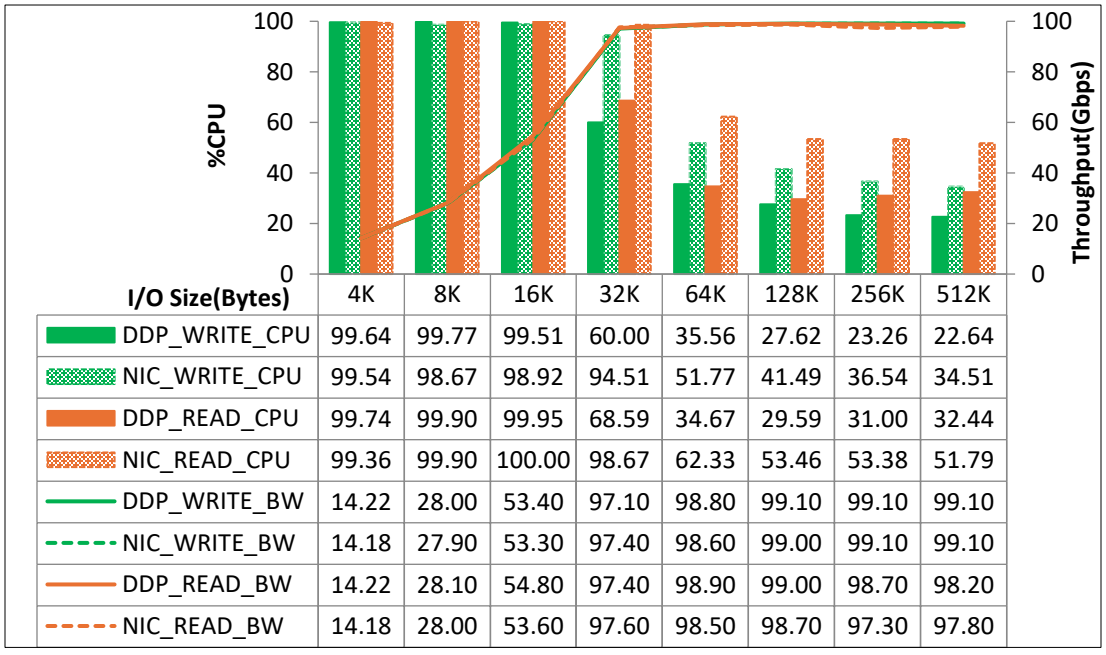
T62100-CR 1x100G MTU 1500



I/O Size(Bytes)	4K	8K	16K	32K	64K	128K	256K	512K
DDP_WRITE_CPU	99.56	98.31	91.92	39.41	28.23	22.95	19.95	19.46
NIC_WRITE_CPU	97.62	93.15	96.56	98.03	78.82	66.72	61.44	58.92
DDP_READ_CPU	99.87	99.82	99.90	69.38	33.08	28.62	29.69	31.10
NIC_READ_CPU	99.79	99.79	99.62	88.51	54.87	47.74	45.64	49.72
DDP_WRITE_BW	14.24	28.10	53.70	84.40	84.50	86.60	87.50	88.40
NIC_WRITE_BW	14.08	27.40	52.90	86.70	89.50	89.80	89.70	88.40
DDP_READ_BW	14.24	28.20	54.90	94.30	94.80	94.80	94.80	94.80
NIC_READ_BW	14.21	28.10	54.40	91.70	92.10	91.20	92.10	91.90

Figure 1 - Throughput & CPU Usage vs. I/O Size

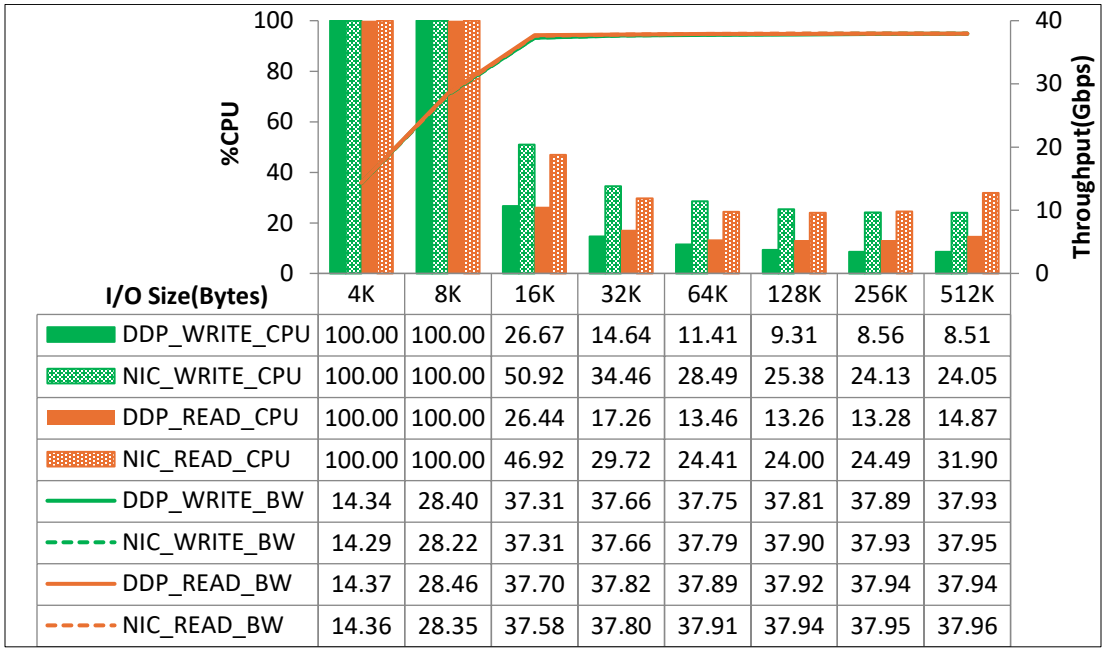
T62100-CR 1x100G MTU 9000



I/O Size(Bytes)	4K	8K	16K	32K	64K	128K	256K	512K
DDP_WRITE_CPU	99.64	99.77	99.51	60.00	35.56	27.62	23.26	22.64
NIC_WRITE_CPU	99.54	98.67	98.92	94.51	51.77	41.49	36.54	34.51
DDP_READ_CPU	99.74	99.90	99.95	68.59	34.67	29.59	31.00	32.44
NIC_READ_CPU	99.36	99.90	100.00	98.67	62.33	53.46	53.38	51.79
DDP_WRITE_BW	14.22	28.00	53.40	97.10	98.80	99.10	99.10	99.10
NIC_WRITE_BW	14.18	27.90	53.30	97.40	98.60	99.00	99.10	99.10
DDP_READ_BW	14.22	28.10	54.80	97.40	98.90	99.00	98.70	98.20
NIC_READ_BW	14.18	28.00	53.60	97.60	98.50	98.70	97.30	97.80

Figure 2 - Throughput & CPU Usage vs. I/O Size

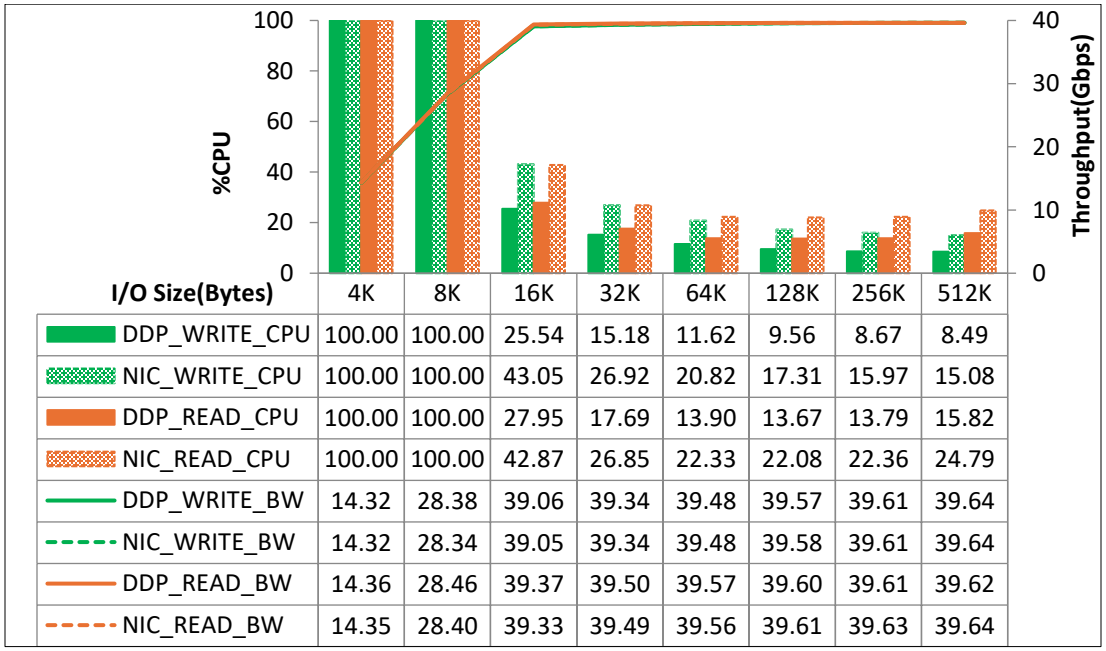
T540-LP-CR 4x10G MTU 1500



I/O Size(Bytes)	4K	8K	16K	32K	64K	128K	256K	512K
DDP_WRITE_CPU	100.00	100.00	26.67	14.64	11.41	9.31	8.56	8.51
NIC_WRITE_CPU	100.00	100.00	50.92	34.46	28.49	25.38	24.13	24.05
DDP_READ_CPU	100.00	100.00	26.44	17.26	13.46	13.26	13.28	14.87
NIC_READ_CPU	100.00	100.00	46.92	29.72	24.41	24.00	24.49	31.90
DDP_WRITE_BW	14.34	28.40	37.31	37.66	37.75	37.81	37.89	37.93
NIC_WRITE_BW	14.29	28.22	37.31	37.66	37.79	37.90	37.93	37.95
DDP_READ_BW	14.37	28.46	37.70	37.82	37.89	37.92	37.94	37.94
NIC_READ_BW	14.36	28.35	37.58	37.80	37.91	37.94	37.95	37.96

Figure 3 - Throughput & CPU Usage vs. I/O Size

T540-LP-CR 4x10G MTU 9000



I/O Size(Bytes)	4K	8K	16K	32K	64K	128K	256K	512K
DDP_WRITE_CPU	100.00	100.00	25.54	15.18	11.62	9.56	8.67	8.49
NIC_WRITE_CPU	100.00	100.00	43.05	26.92	20.82	17.31	15.97	15.08
DDP_READ_CPU	100.00	100.00	27.95	17.69	13.90	13.67	13.79	15.82
NIC_READ_CPU	100.00	100.00	42.87	26.85	22.33	22.08	22.36	24.79
DDP_WRITE_BW	14.32	28.38	39.06	39.34	39.48	39.57	39.61	39.64
NIC_WRITE_BW	14.32	28.34	39.05	39.34	39.48	39.58	39.61	39.64
DDP_READ_BW	14.36	28.46	39.37	39.50	39.57	39.60	39.61	39.62
NIC_READ_BW	14.35	28.40	39.33	39.49	39.56	39.61	39.63	39.64

Figure 4 - Throughput & CPU Usage vs. I/O Size

Results Summary

- T62100-CR provides CPU savings of 60% - 40% for WRITE and 20% for READ at MTU 1500.
- T62100-CR provides CPU savings of 35% - 12% for WRITE and 30% - 20% for READ at MTU 9000.
- T540-LP-CR provides CPU savings of 24% - 15% with WRITE and 20% - 10% for READ at MTU 1500.
- T540-LP-CR provides CPU savings of 18% - 7% with WRITE and 15% - 9% for READ at MTU 9000.
- Consistent line-rate throughput is delivered in both L2 NIC and TCP Offload modes by both the adapters.

The TCP Offload solution provides significant CPU savings, indicative of a more efficient data processing path. In addition, DDP does not pollute the CPU caches unnecessarily and thus saves bandwidth on the memory bus.

Critically, a TCP Offload based solution provides a lower latency and lower latency variation for NFS applications. This is especially important currently where many filers front-end flash based arrays, where latency is critical. Further, having the TOE+DDP activity be accomplished by the adapter, will allow the application to continue to reside in processor cache and not be perturbed by the need to retransmit any packets on the wire, thus reducing wire packet jitter and improving the CPU utilization and efficiency of the filer.

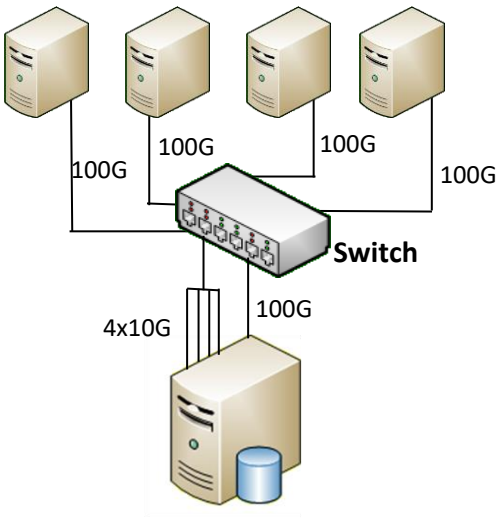
A major benefit of the reduced CPU utilization enabled by Chelsio TCP Offload is radically reduced Capex for NFS/TCP server deployments. Specifically, ~30% of a typical ~$10k NFS filter head cost is saved by using Chelsio T6 TCP Offload which translates to ~$3K+ savings per head for 100Gb operation. If the filer happens to not have enough CPU% headroom, then the savings presented by a TCP Offload capable NIC are specially more valuable than above. And of course, the use of offload will come with an improvement in IOPs and latency of the filer. Further, the Capex impact of the efficiency of protocol offload offered by Chelsio Terminator architecture will be magnified as data centers transition to 200Gbps and 400Gbps speeds.

At lower I/O sizes, there is no significant data bandwidth impact due to filesystem limitations. The CPU is fully utilized handling the tmpfs and NFS spin-locks which does not lend itself to offload.

Test Configuration

The setup consists of an NFSv4 server and 4 clients connected through a 100GbE switch using a single 100G port on each system. In case of T540-LP-CR, all 4 ports are connected to the switch using a 40G -> 4x10G splitter cable. The NFS Server is configured with FreeBSD inbox drivers for Chelsio adapters. The Chelsio Unified Wire driver for Linux v3.18.0.2 is installed on the Linux client machines. MTU of 1500 and 9000 are used on all the ports.

For T62100-CR test, the Server exposes 2 shares. 2 client machines are used and access 1 share each using 8 connections. For T540-LP-CR test, the Server exposes 4 shares. 4 client machines are used and access 1 share each using 4 connections.



- Supermicro X10DRH clients
- 2 Intel Xeon CPUs E5-2650 v3 10-core @ 2.30GHz (HT disabled)
- 128 GB RAM
- RHEL 8.5
- Chelsio T62100-CR

- Supermicro X10DRG-Q Server
- 2 Intel Xeon CPUs E5-2687W v4 12-core @ 3.00GHz (HT disabled)
- 128 GB RAM
- FreeBSD 15.0-CURRENT
- Chelsio T62100-CR/T540-LP-CR

Figure 5 – Test setup

Setup Configuration

NFS Server Configuration

i. Disable virtualization, c-state technology, VT-d, Intel I/O AT, SR-IOV in system BIOS.
ii. Install FreeBSD 15.0-CURRENT OS.
iii. Compile and install the latest changeset of the FBSD head repo. Reboot the machine into the installed kernel.
iv. Configure the below settings.

```
[root@server~]# cat /boot/loader.conf
vfs.maxbcachebuf=1048576
hw.cxgbe.ntxq="-16"
hw.cxgbe.nrxq="-16"
hw.cxgbe.nofldtxq="-16"
hw.cxgbe.nofldrxq="-16"
kern.hwpmc.nbuffers_pcpu="128"
kern.hwpmc.nsamples="16384"

[root@server~]# cat /etc/sysctl.conf
dev.t6nex.0.toe.rx_coalesce=0
dev.t5nex.0.toe.rx_coalesce=0
kern.ipc.maxsockbuf=6291456
vfs.nfsd.async=1

[root@server~]# cat /etc/rc.conf
kld_list="filemon dtraceall hwpmc cpuctl ksyms vmm nmdm"
kld_list="$kld_list t4_tom"
ifconfig_cc0="inet 102.100.1.145/24"
ifconfig_cxl0="inet 102.10.1.145/24"
ifconfig_cxl1="inet 102.10.2.145/24"
ifconfig_cxl2="inet 102.10.3.145/24"
ifconfig_cxl3="inet 102.10.4.145/24"
nfs_server_enable="YES"
nfsv4_server_only="YES"
nfs_server_maxio="1048576"
```

v. Create 4 tmpfs shares, each of size 16GB.

```
[root@server~]#  mkdir  -p  /shared/share1  /shared/share2  /shared/share3
/shared/share4
[root@server~]# cat /etc/fstab
# Device        Mountpoint      FStype  Options Dump    Pass#
tmpfs   /shared/share1  tmpfs   rw,size=16g     0       0
tmpfs   /shared/share2  tmpfs   rw,size=16g     0       0
tmpfs   /shared/share3  tmpfs   rw,size=16g     0       0
tmpfs   /shared/share4  tmpfs   rw,size=16g     0       0
```

vi. Export the shares.

```
[root@server~]# cat /etc/exports
V4: /shared
/shared/share1 -mapall=root
/shared/share2 -mapall=root
/shared/share3 -mapall=root
/shared/share4 -mapall=root
```

vii. Configure the T5/T6 interface(s) with L2 NIC/TOE DDP modes with the required MTU.

L2 NIC:

```
[root@server~]# ifconfig cc0 up
[root@server~]# for i in {0..3}; do ifconfig cxl$i up; done
```

TOE DDP:

```
[root@server~]# ifconfig cc0 toe
[root@server~]# sysctl dev.t6nex.0.toe.ddp=1

[root@server~]# for i in {0..3}; do ifconfig cxl$i toe; done
[root@server~]# sysctl dev.t5nex.0.toe.ddp=1
```

NOTE: Restart the NFS server service after configuring the different modes.

```
[root@server~]# service nfsd restart
```

Clients Configuration

i. Disable virtualization, c-state technology, VT-d, Intel I/O AT, SR-IOV in system BIOS.
ii. Compile and install the Chelsio Unified Wire package and reboot the machine.

```
[root@client~]# cd ChelsioUwire-3.18.0.2
[root@client~]# make install
[root@client~]# reboot
```

iii. Set cpupower governor to performance.

```
[root@client~]# cpupower frequency-set --governor performance
```

iv. Set the below tuned-adm profile for BW/IOPs test.

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

v. Load the Chelsio NIC driver (*cxgb4*) and bring up interface with IPv4 address and MTU.

```
[root@client~]# modprobe cxgb4
[root@client~]# ifconfig ethX <IPv4 address> up
```

vi. CPU affinity was set.

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

vii. The clients mount the exported shares.

```
T62100-CR:
[root@client1~]# mount -o nconnect=8 102.100.1.145:/share1 /mnt/nfs
[root@client2~]# mount -o nconnect=8 102.100.1.145:/share2 /mnt/nfs

T540-LP-CR:
[root@client1~]# mount -o nconnect=4 102.10.1.145:/share1 /mnt/nfs
[root@client2~]# mount -o nconnect=4 102.10.2.145:/share2 /mnt/nfs
[root@client3~]# mount -o nconnect=4 102.10.3.145:/share3 /mnt/nfs
[root@client4~]# mount -o nconnect=4 102.10.4.145:/share4 /mnt/nfs

[root@client1~]# mount -v |grep -i 102.100.1.145
102.100.1.145:/share1 on /mnt/nfs type nfs4
(rw,relatime,vers=4.2,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,
nconnect=8,timeo=600,retrans=2,sec=sys,clientaddr=102.100.1.1,local_lock=non
e,addr=102.100.1.145)
```

viii. *fio* tool (v3.19) was run from all the Clients concurrently on the mounted shares.

```
[root@client1 .. client4~]# fio --rw=randread/randwrite --filesize=500MB --
name=random --norandommap --group_reporting --exitall --fsync_on_close=1 --
invalidate=1 --randrepeat=0 --direct=1 --directory=/mnt/nfs --time_based --
runtime=60 --iodepth=16 --numjobs=8 --ioengine=libaio --unit_base=1 --
bs=${blk_size} --kb_base=1000 --ramp_time=5
```

Conclusion

This paper compares the performance of NFS/TCP Server with and without TCP Offload using Chelsio's T62100-CR and T540-LP-CR Unified Wire Adapters in FreeBSD. It illustrates the benefits of using Chelsio's TCP Offload to provide exceptional CPU savings compared to an L2 NIC. When TOE DDP is in use, T6 delivers up to 60% reduction in CPU usage compared to a typical L2 NIC when using 1500B packets. The freed-up CPU processing cycles can be used for running other important applications and maximizes the data center efficiency. As data centers become saturated with information, the need to offload tasks from server's CPU is more important now than ever.

Related Links

The True Cost of Non-Offloaded NICs
The Chelsio Terminator 6 ASIC
100G Kernel and User Space NVMe/TCP Using Chelsio Offload
Industry's First 100G Offload with FreeBSD
FreeBSD 40GbE TOE Performance