100G NGINX Offload & Encryption

Chelsio T6 vs. Mellanox ConnectX-6 Dx

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. Additionally, the Inline TLS/SSL offload capability offered by Chelsio T6 adapters is uniquely capable of producing record breaking TLS/SSL performance. Chelsio Inline TLS/SSL offload solution supports TCP/IP and TLS/SSL processing in a cut-through fashion to achieve optimal bandwidth and latency. The Chelsio adapters are designed specifically to perform computationally intensive network and cryptographic operations more efficiently than general-purpose CPUs. Servers with system load comprising of network and cryptographic operations see great savings in CPU Utilization by offloading these operations to the Chelsio T6 adapter.

This paper presents a performance comparison of 100G Chelsio T6 Inline TLS/SSL offload vs. Mellanox ConnectX-6 Dx kTLS offload with a real-world application, NGINX Plus web server running in Linux. It establishes the fact that using Chelsio T6 to offload both the network and cryptographic operations, the NGINX web server consumes significantly less CPU with consistent performance which translates to Chelsio T6 enabling radically reduced Capex for NGINX server deployments.

Test Results

The following graph presents the %CPU and throughput results of NGINX Plus web server, using Chelsio T6 Inline TLS/SSL offload and Mellanox ConnectX-6 Dx kTLS offload. The numbers are collected using *wrk* tool with connections ranging from 1000 to 10000.

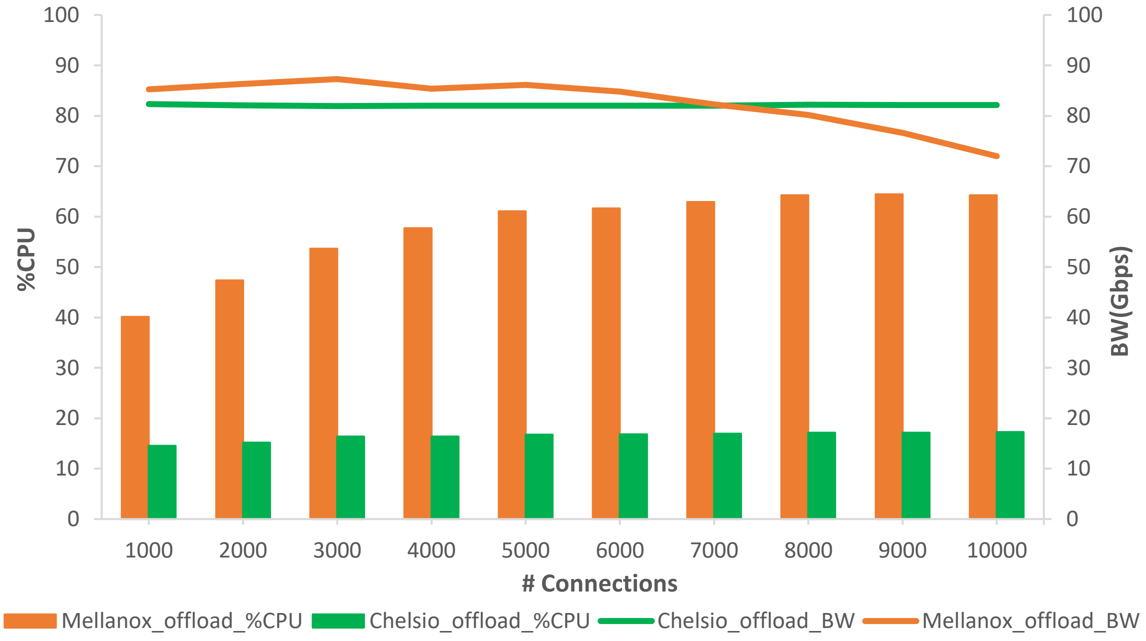


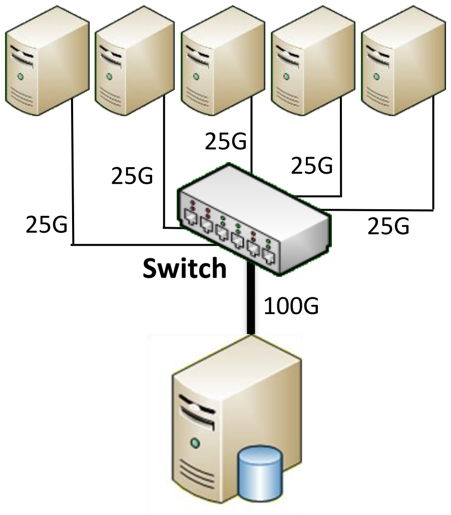
Figure 1 - Throughput & CPU Usage vs. # Connections

The Chelsio Inline TLS/SSL offload solution shows 15% to 17% CPU utilization versus 40% to 64% CPU utilization for Mellanox ConnectX-6 for 1000 to 10000 connections which translated to Chelsio T6 providing up to 70% CPU savings compared to the Mellanox Solution.

A major benefit of the reduced CPU utilization enabled by Chelsio T6 is radically reduced Capex for NGINX server deployments. Specifically, 45% of a typical $5k server cost is saved by using Chelsio T6 instead of Mellanox ConnectX-6 which translates to $2K+ savings per NGINX server.

Considering that data centers may (depending on scalability requirements) typically deploy tens to hundreds to thousands of NGINX servers, such cost savings enabled by T6 can make a make a major impact on overall Capex. 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 speed.

Test Configuration



- Supermicro X10DRH clients
- 2 Intel Xeon CPUs E5-2650 v3 10-core @ 2.30GHz (HT disabled)
- 128 GB RAM
- RHEL 9.1
- Chelsio T6225-CR

- Supermicro X10DRG-Q Web Server
- 2 Intel Xeon CPUs E5-2687W v4 12-core @ 3.00GHz (HT disabled)
- 128 GB RAM
- RHEL9.1
- Chelsio T62100-CR
- Mellanox ConnectX-6 Dx

Figure 2 – Test setup

The setup consists of a server connecting to the Switch with a single 100G port (Chelsio T62100-CR or Mellanox ConnectX-6 Dx). 5 client machines are connected to the same switch, each using a single 25G port of Chelsio T6225-CR. The Chelsio Unified Wire driver for Linux v3.18.0.0 is installed on all machines. Mellanox ConnectX-6 Dx is configured with inbox drivers. A standard MTU of 1500B is used on all the ports.

Each client establishes 200 to 2000 connections using *wrk* tool to download the files from the NGINX Plus web server.

Setup Configuration

General Configuration

Execute the below steps on the server and all client machines.

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@host~]# cd ChelsioUwire-3.18.0.0
[root@host~]# make toe_install
[root@host~]# reboot
```

iii. Set cpupower governor to performance.

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

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

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

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

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

vi. Configure the below sysctls for running higher number of connections.

```
[root@host~]# sysctl net.core.somaxconn=40960
[root@host~]# sysctl net.ipv4.tcp_tw_reuse=1
```

Server Configuration

i. Install NGINX plus by following the steps from NGINX Docs. Verify the nginx binary version.

```
[root@server~]# nginx -v
nginx version: nginx/1.23.4 (nginx-plus-r29)
```

ii. Configure the web server by updating required settings in */etc/nginx/nginx.conf*

```
#user  nobody;
worker_processes  auto;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    sendfile        on;
    ssl_conf_command Options KTLS;
    keepalive_timeout  65;

    server {
        listen       443 ssl backlog=20480;
        server_name  0.0.0.0;

        keepalive_timeout  3600;
        keepalive_requests 100000;

        ssl_certificate      /root/server.crt;
        ssl_certificate_key  /root/server.key;

# SSL Settings
# ##
        ssl_protocols TLSv1.2; # Dropping TLSv1 SSLv3, ref: POODLE
        ssl_ciphers AES128-GCM-SHA256;
        ssl_prefer_server_ciphers on;

        location /2MiB/ {
             sendfile on;
             root /nginx;
                 }
        }
}
```

iii. Create a data file for the clients to download.

```
[root@server~]# mkdir -p /nginx/2MiB/
[root@server~]# dd if=/dev/urandom of=/nginx/2MiB/1_1.txt bs=1024 count=2000
oflag=direct iflag=fullblock status=progress
```

iv. Configure the server with Chelsio and Mellanox offloads.

```
Chelsio Inline TLS/SSL offload:
[root@server~]# modprobe -v t4_tom
[root@ server~]# cat nginx_tls_cop.rule
src or dst port 443 => offload tls mss 32 bind random !nagle
all => offload
[root@server~]# cop -d -o nginx_tls_cop.policy nginx_tls_cop.rule
[root@server~]# cxgbtool ethX policy nginx_tls_cop.policy
[root@server~]# t4_perftune.sh -s -n -Q ofld

Mellanox kTLS offload:
[root@server~]# modprobe -v mlx5_core
[root@server~]# ifconfig ethX <IPv4 address> up
[root@server~]# set_irq_affinity_bynode.sh 0 ethX
[root@server~]# ethtool -K ethX tls-hw-tx-offload on
[root@server~]# ethtool -K ethX tls-hw-rx-offload on
```

v. Start the NGINX plus server.

```
[root@server~]# nginx
```

NOTE: Restart the web server after configuring the different modes specified in step iv.

Clients Configuration

i. CPU affinity was set.

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

ii. *wrk* tool (v4.2.0) was run from all the 5 Clients to download the file.

```
$inst = 10, 20 .. 100
```

```
[root@client1~]# for i in $(seq 1 ${inst}) ; do /root/wrk-4.2.0/wrk -t 1 -c
20 -d 100s https://<Server IP>/2MiB/1_1.txt & done
..
[root@client5~]# for i in $(seq 1 ${inst}) ; do /root/wrk-4.2.0/wrk -t 1 -c
20 -d 100s https://<Server IP>/2MiB/1_1.txt & done
```

Conclusion

This paper compared the performance of Chelsio's T6 Inline TLS/SSL offload solution with Mellanox ConnectX-6 Dx kTLS offload solution in Linux. With a consistent throughput and considerable CPU savings (even with 10000 connections), Chelsio's solutions proves to be the best choice for users looking for the highest level of data security and integrity, without compromising performance. T6 is currently the only secure engine capable of full TCP/IP Offload at 100Gbps. The introduction of integrated encryption within a NIC price and power envelope should further the migration towards secure cloud networks and storage.

Related Links

25G NGINX Offload Performance
NGINX Applications Offload
The True Cost of Non-Offloaded NICs
Concurrent Offload & Encryption at 100GbE
Chelsio Terminator 6 ASIC 100GE Crypto Offload
The Chelsio Terminator 6 ASIC