NVMe/TCP & iSCSI JBOF with T7

Using Chelsio T7, ASMedia PCIe Switch & Samsung SSD

Overview

The Terminator 7 (T7) ASIC from Chelsio Communications, Inc. is a seventh generation, high-performance 1/10/25/40/50/100/200/400 Gbps, Smart NIC Data Processing Unit (DPU) that offers offload support for a wide range of Storage (NVMe/TCP, NVMe-oF, iSCSI, iSER, S2D, SMB), Network (TCP/IP, UDP/IP, RDMA - iWARP and RoCEv2), Virtualization (SR-IOV, VMQ, VXLAN, NVGRE, Geneve), Crypto (IPsec, TLS/SSL), classification and filtering, and Traffic Management protocols. T7 has an embedded general-purpose ARM processor, fully capable of supporting all the functions of the host server processor. Chelsio Smart NICs unburden communication responsibilities and processing overhead from servers and storage systems resulting in a significant saving of server CPU cycles. This can permit radically reduced system cost by enabling the use of a less expensive CPU or the freed-up CPU can be used for running other applications or workloads.

T7 is a high-performance server or embedded ASIC, fully capable of supporting all the functions of a typical system on a single chip. T7 can be configured as a regular End Point (EP), which can be managed by the server as a traditional NIC or offload adapter. T7 has dual quad-core 1.8GHz, A72 ARM processors which can be used for any additional assistance or higher level processing of the data that is tunneled or offloaded. The ARM system on T7 comes pre-loaded with a Linux kernel. It can be used to run full-offload operations like iSCSI, NVMe/TCP, and NVMe-oF etc. The users can also deploy their custom software stack on the ARM system.

T7 can be configured as a Root Complex (RC), making it the host for other end point devices like PCIe SSDs. The ARM system, embedded in T7 can configure and manage these SSDs. T7 will now act as a JBOF (Just a Bunch Of Flash) storage controller, moving the data directly to the SSDs, without involving the server CPU.

This paper describes the T7's capability to be used as a JBOF and demonstrates it with the T7 Emulation platform, ASMedia PCIe Switch and Samsung SSD. An iSCSI and NVMe/TCP target are configured on the T7 ARM system, bypassing the server. The Initiator connects to the target over the T7 ethernet ports and accesses the SSD.

ASMedia PCIe product ASM2824, is a low latency, low power, and low cost packet switch which will enable users to build up various high-speed IO systems, system storage, or communication platforms.

The Demonstration

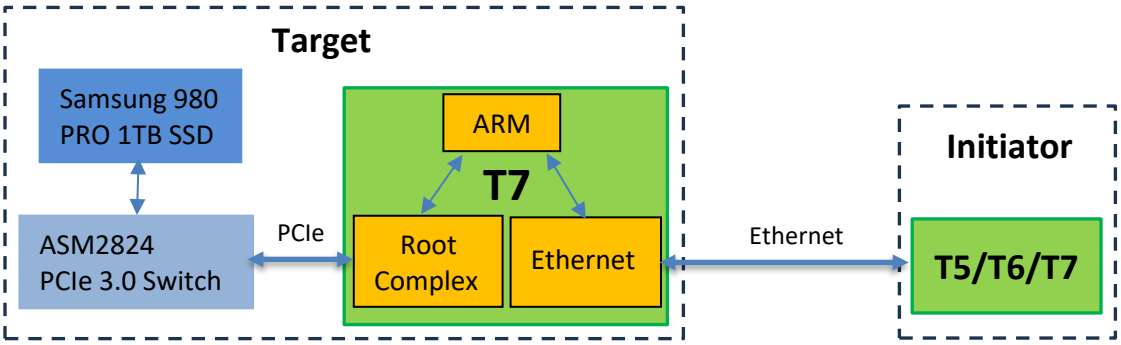

Figure 1 – JBOF with T7 ARM, ASMedia PCIe Switch and Samsung SSD

The test setup consists of a server with the T7 emulation platform configured in RC mode. The ASMedia ASM2824 PCIe 3.0 Switch, with a Samsung 980 PRO 1 TB NVMe SSD connects to the T7 Root Complex. The T7 ARM system detects the Samsung SSD and configures an LIO iSCSI target and NVMe/TCP Target, one at a time.

The Initiator, with a T6 adapter in iSCSI or NVMe/TCP modes, connects to the target running on the T7 ARM system. The initiator connects directly (no switch) to the T7 Ethernet port. A standard MTU of 1500 is used on the ports under test.

iSCSI Configuration

Target

i. Configure the T7 emulation platform with the required RC images and firmware.
ii. Connect to the T7 ARM system.
iii. Verify that the NVMe SSD is detected.

```
# lspci
00:00.0 PCI bridge: Chelsio Communications Inc Device d000
01:00.0 PCI bridge: ASMedia Technology Inc. ASM2824 PCIe Gen3 Packet Switch
(rev 01)
02:00.0 PCI bridge: ASMedia Technology Inc. ASM2824 PCIe Gen3 Packet Switch
(rev 01)
02:04.0 PCI bridge: ASMedia Technology Inc. ASM2824 PCIe Gen3 Packet Switch
(rev 01)
02:08.0 PCI bridge: ASMedia Technology Inc. ASM2824 PCIe Gen3 Packet Switch
(rev 01)
02:0c.0 PCI bridge: ASMedia Technology Inc. ASM2824 PCIe Gen3 Packet Switch
(rev 01)
03:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD
Controller SM981/PM981/PM983
```

iv. Load the Chelsio NIC driver (*cxgb4*) and bring up the T7 Ethernet port with an IPv4 address.

```
# insmod cxgb4.ko
# ifconfig ethX 102.55.55.70/24 up
```

v. Load the LIO Target driver.

```
# insmod cxgbit.ko
```

vi. Configure the target with Samsung SSD using the below script.

```
# sh iscsi_target.sh /dev/nvme1n1p1 102.55.55.70
Target iqn.2003-01.org.linux-iscsi.buildroot:nvme1n1p1, portal 0.0.0.0:3260
has been created

# cat iscsi_target.sh

print_usage() {
    cat <<EOF
Usage: $(basename $0) [-p PORTAL] DEVICE|FILE
Export a block device or a file as an iSCSI target with a single LUN
EOF
}

die() {
    echo $1
    exit 1
}

while getopts "hp:" arg; do
    case $arg in
        h) print_usage; exit 0;;
        p) PORTAL=${OPTARG};;
    esac
done
shift $(($OPTIND - 1))

DEVICE=$1
[ -n "$DEVICE" ] || die "Missing device or file argument"
[ -b $DEVICE -o -f $DEVICE ] || die "Invalid device or file: ${DEVICE}"
IQN="iqn.2003-01.org.linux-iscsi.$(hostname):$(basename $DEVICE)"
[ -n "$PORTAL" ] || PORTAL="0.0.0.0:3260"

CONFIGFS=/sys/kernel/config
CORE_DIR=$CONFIGFS/target/core
ISCSI_DIR=$CONFIGFS/target/iscsi

# Load the target modules and mount the config file system
lsmod | grep -q configfs || modprobe configfs
lsmod | grep -q target_core_mod || modprobe target_core_mod
mount | grep -q ^configfs || mount -t configfs none $CONFIGFS
mkdir -p $ISCSI_DIR

# Create a backstore
if [ -b $DEVICE ]; then
    BACKSTORE_DIR=$CORE_DIR/iblock_0/data
    mkdir -p $BACKSTORE_DIR
    echo "udev_path=${DEVICE}" > $BACKSTORE_DIR/control
else
    BACKSTORE_DIR=$CORE_DIR/fileio_0/data
    mkdir -p $BACKSTORE_DIR
    DEVICE_SIZE=$(du -b $DEVICE | cut -f1)
    echo "fd_dev_name=${DEVICE}" > $BACKSTORE_DIR/control
    echo "fd_dev_size=${DEVICE_SIZE}" > $BACKSTORE_DIR/control
```

```
    echo 1 > $BACKSTORE_DIR/attrib/emulate_write_cache
fi
echo 1 > $BACKSTORE_DIR/enable

# Create an iSCSI target and a target portal group (TPG)
mkdir $ISCSI_DIR/$IQN
mkdir $ISCSI_DIR/$IQN/tpgt_1/

# Create a LUN
mkdir $ISCSI_DIR/$IQN/tpgt_1/lun/lun_0
ln -s $BACKSTORE_DIR $ISCSI_DIR/$IQN/tpgt_1/lun/lun_0/data
echo 1 > $ISCSI_DIR/$IQN/tpgt_1/enable

# Create a network portal
mkdir $ISCSI_DIR/$IQN/tpgt_1/np/$PORTAL

# Disable authentication
echo 0 > $ISCSI_DIR/$IQN/tpgt_1/attrib/authentication
echo 1 > $ISCSI_DIR/$IQN/tpgt_1/attrib/generate_node_acls

# Allow write access for non authenticated initiators
echo 0 > $ISCSI_DIR/$IQN/tpgt_1/attrib/demo_mode_write_protect

echo "Target ${IQN}, portal ${PORTAL} has been created"
```

Initiator

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

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

ii. Login to the target.

```
[root@initiator~]# iscsiadm -m discovery -t st -p 102.55.55.70:3260 -I
default -l

102.55.55.70:3260,1 iqn.2003-01.org.linux-iscsi.buildroot:nvme1n1p1
Logging in to [iface: default, target: iqn.2003-01.org.linux-
iscsi.buildroot:nvme1n1p1, portal: 102.55.55.70,3260]
Login to [iface: default, target: iqn.2003-01.org.linux-
iscsi.buildroot:nvme1n1p1, portal: 102.55.55.70,3260] successful.
```

iii. Format the disk and mount it.

```
[root@initiator~]# lsscsi
[0:0:0:0]    disk    ATA      SanDisk SDSSDA12 80RL  /dev/sda
[1:0:0:0]    disk    ATA      Samsung SSD 840  CB6Q  /dev/sdb
[4:0:0:0]    disk    ATA      OCZ-VERTEX3      2.22  /dev/sdc
[6:0:0:0]    disk    LIO-ORG  IBLOCK           4.0   /dev/sdd

[root@initiator~]# mkfs.ext4 /dev/sdd
mke2fs 1.45.6 (20-Mar-2020)
/dev/sdd contains a ext3 file system
        last mounted on /mnt on Thu Nov 30 03:49:32 2023
Proceed anyway? (y,N) y
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: c326f769-8fcd-46ea-8057-843452f7f471
```

```
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[root@initiator~]# mount /dev/sdd /mnt/iscsi0/
```

iv. *iozone* tool was run on the mounted disk.

```
[root@initiator~]# cd /mnt/iscsi0/
[root@initiator~]# iozone -a -d -+I
       Iozone: Performance Test of File I/O
                Version $Revision: 3.493 $
                Compiled for 64 bit mode.
                Build: linux-AMD64

...
        Run began: Thu Nov 30 08:12:33 2023

        Auto Mode
        Command line used: iozone -a -d -+I
        Output is in kBytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 kBytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
                                                                    random
random     bkwd    record    stride
              kB  reclen    write  rewrite    read    reread    read
write     read   rewrite      read   fwrite frewrite    fread  freread
              64       4   901377  2923952  6271021  4897948  4643754
2561267  3203069   3203069   3791156  2772930  3057153  4274062  7100397
              64       8  1143223  3791156 10402178  7100397  7100397
3363612  4564786   4274062   6421025  3541098  3588436  7100397 ..
```

NVMe/TCP Configuration

Target

i. Configure the T7 emulation platform with the required RC images and firmware.
ii. Connect to the ARM system.
iii. Verify that the NVMe SSD is detected.

```
# nvme list
Node             SN                   Model
Namespace Usage                      Format           FW Rev
---------------- -------------------- ----------------------------------------
-- --------- -------------------------- ---------------- --------
/dev/nvme0n1     S5P2NU0WA02411Y      Samsung SSD 980 PRO 1TB
1          15.93  GB /   1.00  TB    512   B +  0 B   5B2QGXA7

# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
nvme0n1     259:0    0 931.5G  0 disk
`-nvme0n1p1 259:1    0     1G  0 part
```

iv. Load the Chelsio NIC driver (*cxgb4*) and bring up the T7 Ethernet port with an IPv4 address.

```
# insmod cxgb4.ko
# ifconfig ethX 102.55.55.70/24 up
```

v. Load the NVMe/TCP PDU Offload Target driver.

```
# insmod cnvmet-tcp.ko
```

vi. Configure the target with Samsung SSD using the below script.

```
# sh nvme_tcp_target.sh /dev/nvme0n1p1 102.55.55.70

# cat nvme_tcp_target.sh

mount -t configfs none /sys/kernel/config
mkdir /sys/kernel/config/nvmet/subsystems/nvme-ssd0
mkdir /sys/kernel/config/nvmet/subsystems/nvme-ssd0/namespaces/1
echo -n $1  >/sys/kernel/config/nvmet/subsystems/nvme-
ssd0/namespaces/1/device_path
echo 1 > /sys/kernel/config/nvmet/subsystems/nvme-ssd0/attr_allow_any_host
echo 1 > /sys/kernel/config/nvmet/subsystems/nvme-ssd0/namespaces/1/enable
mkdir /sys/kernel/config/nvmet/ports/1
echo ipv4 > /sys/kernel/config/nvmet/ports/1/addr_adrfam
echo tcp > /sys/kernel/config/nvmet/ports/1/addr_trtype
echo 4420 > /sys/kernel/config/nvmet/ports/1/addr_trsvcid
echo  $2 > /sys/kernel/config/nvmet/ports/1/addr_traddr
cd /sys/kernel/config/nvmet/ports/1/subsystems/
ln -s /sys/kernel/config/nvmet/subsystems/nvme-ssd0
/sys/kernel/config/nvmet/ports/1/subsystems/nvme-ssd0
```

Initiator

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

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

ii. Login to the target.

```
[root@initiator~]# nvme connect -t tcp --nqn nvme-ssd0 -a 102.55.55.70 -s 4420
```

iii. *fio* tool was run.

```
[root@initiator~]# fio --ioengine=libaio --rw=randwrite --name=randwrite --
size=200m --direct=1 --invalidate=1 --fsync_on_close=1 --group_reporting --
exitall --runtime=20 --time_based --filename=/dev/nvme0n1 --iodepth=4 --
numjobs=8 --bs=4k
     randwrite: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio,
iodepth=4
     ...
     fio-2.1.10
     Starting 8 processes
     Jobs: 8 (f=8): [wwwwwwww] [100.0% done] [0KB/1048KB/0KB /s] [0/262/0 iops]
[eta 00m:00s]
     randwrite: (groupid=0, jobs=8): err= 0: pid=73024: Mon Dec  4 09:11:20
2023
```

```
        write: io=21636KB, bw=1077.7KB/s, iops=269, runt= 20088msec
          slat (usec): min=7, max=71, avg=12.56, stdev= 2.59
          clat (msec): min=8, max=211, avg=118.56, stdev=33.31
           lat (msec): min=8, max=211, avg=118.58, stdev=33.31
          clat percentiles (msec):
           |  1.00th=[   25],  5.00th=[   32], 10.00th=[  101], 20.00th=[  113],
           | 30.00th=[  117], 40.00th=[  119], 50.00th=[  121], 60.00th=[  122],
           | 70.00th=[  126], 80.00th=[  133], 90.00th=[  151], 95.00th=[  180],
           | 99.00th=[  196], 99.50th=[  200], 99.90th=[  208], 99.95th=[  212],
           | 99.99th=[  212]
          bw (KB  /s): min=   82, max=  226, per=12.45%, avg=134.10, stdev=28.08
          lat (msec) : 10=0.24%, 20=0.52%, 50=6.66%, 100=2.57%, 250=90.02%
        cpu          : usr=0.01%, sys=0.04%, ctx=5425, majf=0, minf=62
        IO depths    : 1=0.1%, 2=0.3%, 4=99.6%, 8=0.0%, 16=0.0%, 32=0.0%,
>=64=0.0%
           submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
           complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
           issued    : total=r=0/w=5409/d=0, short=r=0/w=0/d=0
           latency   : target=0, window=0, percentile=100.00%, depth=4

     Run status group 0 (all jobs):
       WRITE: io=21636KB, aggrb=1077KB/s, minb=1077KB/s, maxb=1077KB/s,
mint=20088msec, maxt=20088msec
```

NOTE: These numbers are only preliminary results on the emulation platform. They will further improve with an actual adapter.

Conclusion

The Chelsio T7 solution enables NVMe SSDs to be shared, pooled, and managed more effectively across a low-latency, high-performance network. The T7 ARM can do the complete processing, freeing up significant server CPU resources for application processing. As data centers become saturated with information, the need to offload tasks from server's CPU is more important now than ever.

With concurrent support for offloading multiple network, storage, virtualization, and crypto protocols, and delivering industry-leading performance and efficiency, Chelsio has taken the Unified Wire solution to the next level. All the traffic runs over a single network, rather than building and maintaining multiple networks, resulting in significant acquisition and operational cost savings.

Related Links

T7 Product Brief
iSCSI JBOF Demonstration with T7 and Celestica Nebula G2
NVMe/TCP PDU Offload Demonstration with T7
Chelsio T7 DPU Line Launched for 400G Generation