Description of the detailed issues faced while attempting to run the ESnet setup (https://github.com/esnet/esnet-smartnic-hw) on the OCT testbed, more specifically, on a CloudLab node.

March 22, 2023

Email: msada@hawk.iit.edu

- **CloudLab nodes are running on metal and are accessible via SSH.**
- **lsusb -d 0403:6011**
  It returns nothing as the FPGAs are only connected via PCIe and not JTAG, rendering our whole experiment a deadend.

Steps to run ESnet on CloudLab:

Step 1:

1. **SSH into CloudLab**
2. **Git clone:**

   https://github.com/esnet/xilinx-labtools-docker.git

3. **Git clone & submodule:**

   https://github.com/esnet/smartnic-dpdk-docker

4. **Git clone & submodule:**

   https://github.com/esnet/esnet-smartnic-fw

5. **On node:**
   a. **Apt update & upgrade**
   b. **Follow: https://docs.docker.com/engine/install/ubuntu/**
   c. **Follow: https://docs.docker.com/engine/install/linux-postinstall/**
   d. **Try: docker run hello-world**
   e.

   ```
   mkdir -p ~/.docker/cli-plugins/

   curl -SL
   https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-linux-x86_64 -o
   ~/.docker/cli-plugins/docker-compose

   chmod +x ~/.docker/cli-plugins/docker-compose
   ```

   f. **Try: docker compose version**
   g. **Scp the artifacts and the vivado installer to CloudLab**
      scp -i ~/.ssh/id_rsa hello_world
      ./build_dir.hw.xilinx_u280_xdma_201920_3/vadd.xclbin
      fsada@pc151.cloudlab.umass.edu:~

h. **sudo chown fsada**
    Xilinx_Vivado_Lab_Lin_2022.1_0420_0327.tar.gz
    artifacts.esnet-smartnic-hw.export_hwapi.0.zip
i. **Copy vivado installer to**
    xilinx.../vivado-installer/Xilinx_Vivado_Lab_Lin_2022.1_0420_032
    7.tar.gz
j. **mv artifacts.esnet-smartnic-hw.export_hwapi.0.zip**
    esnet-smartnic-fw/sn-hw/

Step 2:

- **Cd into esnet-smartnic-fw**
- **cd $(git rev-parse --show-toplevel)**
  - **cp example.env .env**
- **Edit .env**
- **cd $(git rev-parse --show-toplevel)**
- **docker compose build**
- **docker compose run --rm sn-fw-pkg**

Step 3:

- **Cd into xilinx...**
- **docker build --pull -t xilinx-labtools-docker:v2022.1-latest .**
- **docker image ls**

Step 4:

- **Cd into smartnic-dpdk-docker**
- **docker build --pull -t smartnic-dpdk-docker:${USER}-dev .**
- **docker image ls**
- **If step 4 fails, do the rest of the steps and come back to 4.**

Step 5:

- lspci -d 10ee
  It returns the FPGAs connected via PCIe.

- lspci -d 10ee: -vv
  Lists FPGA devices and their properties
  Make sure that the memory is enabled
- cat /proc/cmdline
  Displays the kernel boot command line parameters.
  Make sure that:
  BOOT_IMAGE=/boot/vmlinuz-5.4.0-126-generic
  root=/dev/mapper/vg0-root ro default_hugepagesz=1G hugepagesz=1G
  hugepages=32 intel_iommu=on iommu=pt
- Make sure that kernel is 5.4.0 and Ubuntu is 20.04 LTS
  As of the time of writing this document, CloudLab's newest Ubuntu version
  is 18.04 LTS, but you can upgrade it to 20.04 LTS using:
  Install the update-manager-core package by running the command sudo
  apt install update-manager-core.

  Edit the /etc/update-manager/release-upgrades file using your preferred
  text editor (e.g. nano, vim, gedit) and change the value of Prompt to lts.

  Run the command sudo do-release-upgrade -d to start the upgrade
  process. The -d option is used to force the upgrade to the development
  version, which will upgrade to version 20.04 LTS.

- grep -A 3 -B 3 iommu /etc/default/grub
  searches for the "iommu" keyword in the "/etc/default/grub" file and
  displays the 3 lines before and after the match.

  GRUB_TIMEOUT_STYLE=hidden
  GRUB_TIMEOUT=0
  GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
  **GRUB_CMDLINE_LINUX_DEFAULT="default_hugepagesz=1G
  hugepagesz=1G hugepages=32 intel_iommu=on iommu=pt"**
  GRUB_CMDLINE_LINUX=""

  Make sure to modify the bolded line.

- Now reboot:
  sudo reboot
  Or use the CloudLab website to manually reboot the node.

- Now test iommu:
  sudo less /var/log/kern.log
- Now test hugepages:
  grep HugePages /proc/meminfo

Step 6:

- If step 4 fails, that is probably because /var/lib/docker is running out of space.
- Use the steps here:
  https://linuxconfig.org/how-to-move-docker-s-default-var-lib-docker-to-another-directory-on-ubuntu-debian-linux
  To change the directory docker uses from /var/lib/docker to a directory under the directory where you have allocated more space.
  The reason for that is that because CloudLab only allows 16GB, which can be extended in a mounted folder, so we move the docker folder there.