Setting Up the Host (Using VART)

For Edge (DPUCZDX8G)

Use the following steps to set up the host for edge:

1. Download `sdk-2020.1.0.0.sh` from here.
2. Install the cross-compilation system environment.

   ```
   $ ./sdk-2020.1.0.0.sh
   ```

3. Follow the prompts to install. The following figure shows the installation process.

   Note that the `~/petalinux_sdk` path is recommended for installation. Regardless of the path you choose for the installation, make sure the path you choose has re-write permissions. Here, install it under `~/petalinux_sdk`.

4. When the installation is complete, follow the prompts and enter the following command.

   ```
   $ bash ~/petalinux_sdk/environment-setup-aarch64-xilinx-linux
   ```

   Note that if you close the current terminal, you need to re-execute the above instructions in the new terminal.

5. Download the `vitis_ai_2020.1-r1.2.0.tar.gz` from here and install it to the petalinux system.

   ```
   $ tar -xzvf vitis_ai_2020.1-r1.2.0.tar.gz -C ~/petalinux_sdk/sysroots/aarch64-xilinx-linux
   ```

6. Cross compile the sample, take `resnet50` as an example.

   ```
   $ cd Vitis-AI/VART/samples/resnet50
   $ bash -x build.sh
   ```

   If the compilation process does not report any error and the executable file `resnet50` is generated, the host environment is installed correctly.

For Cloud (DPUCAHX8H)

Use the following steps to set up the host for cloud. These steps apply to U50, U50LV, and U280 cards.

1. Start the docker container. After the docker image is loaded and running, the Vitis AI runtime is automatically installed in the docker system.
2. Download the xclbin files from here. Untar it, choose the Alveo card and install it. Take `U50` as an example.

   ```
   $ tar -xzvf alveo_xclbin-1.2.1.tar.gz
   $ cd alveo_xclbin-1.2.1/U50/6E300M
   $ sudo cp dpu.xclbin hbm_address_assignment.txt /usr/lib
   ```

3. If there are more than one card installed on the server and you want to specify some cards to run the program, you can set `XLNX_ENABLE_DEVICES` to achieve this function. The following is the usage of `XLNX_ENABLE_DEVICES`.

   - `export XLNX_ENABLE_DEVICES=0` --only use device 0 for DPU
   - `export XLNX_ENABLE_DEVICES=0,1,3` --select device 0, device 1 and device 3 to be used for DPU
   - If you do not set this environment variable, use all devices for DPU by default.