DNNDK AMI User Guide

UG1351 (v1.0) April 11, 2019

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
4/11/2019 Version 1.0	
Initial Xilinx release.	N/A

XILINX

Table of Contents

Chapter 1: Overview

DNNDK Amazon Machine Image (AMI) provides the information required to launch an AWS instance with the DNNDK X86 host running environment. Using this AWS instance, you can quickly start Xilinx Edge AI tutorials without deploying your own host machine.

You can execute Deep Compression Tool (DECENT) and Deep Neural Network Compiler (DNNC) on this AWS instance, then cross compile the DNNC output model with the deployment code, and get the final executable binary file for the evaluation board. Finally, you only need to copy the binary file to the evaluation board. After that you can execute it on the evaluation board and see the effect.

For more information, refer to the Xilinx Edge AI Developer Hub web page:

https://www.xilinx.com/products/design-tools/ai-inference/ai-developer-hub.html#edge

DNNDK

The Deep Neural Network Development Kit (DNNDK) is a Xilinx AI Solution consisting of an Edge AI Platform.

If you are not familiar with DNNDK, refer to DNNDK User Guide (UG1327).

Preparing Your Local Machine

Although it is not necessary to deploy your own X86 host machine, you need a local host machine to connect the AWS instance. This machine can be a Windows, Linux, or iOS operation system. If you use the Windows OS, you can install MobaXterm to execute the `ssh` command.

If the evaluation board is running in standalone mode, you can use the evaluation board to connect AWS directly.

Setting Up the Evaluation Board

1. Prepare an evaluation board supported by DNNDK v2.08. The supported evaluation boards are:
   - ZCU102
   - ZCU104
   - Ultra96
2. Download the evaluation board image from the Edge AI Resources page: https://www.xilinx.com/products/design-tools/ai-inference/ai-developer-hub.html#edge. For example, for the ZCU102 evaluation board, download 2018-12-04-zcu102-desktop-stretch.img.zip.
3. Flash the OS Image to the SD Card.
   For information about flashing the OS image, refer to DNNDK User Guide (UG1327).
4. Execute the following command to configure the IP address: `vim /etc/network/interfaces.`
5. Reboot the board.

6. Copy the tar file (such as xlnx_dnndk_v2.08_190201.tar.gz) to the evaluation board, and execute the following commands:

```
cd ~
tar -xzvf ./xlnx_dnndk_v2.08_190201.tar.gz
scp -r xilinx_dnndk_v2.08/ZCU102 ~/
rm xilinx_dnndk_v2.08 -r
cd ZCU102
sudo ./install.sh
```

Applying an AWS Instance

You must apply an AWS instance, as follows:

1. Set up a p2.xlarge instance using the Xilinx DNNDK AMI v2.08 according the Amazon AWS documents.
2. Get a *.pem file to connect to the AWS instance. For example, the name of this *.pem file is `aws-test.pem`.
3. Start your AWS instance according Amazon AWS documents, then you can get an IP address.
4. Use the following command to log in to the AWS instance:

```
ssh -i ./aws-test.pem ubuntu@<ip-address>
```

Setting Up the AWS Instance

This AWS instance is equipped with NVIDIA GPU Card, so you must install the corresponding CUDA and cuDNN libraries.

Installing CUDA

Log in to the AWS instance, and use the following commands to install CUDA 8.0:

```
mkdir ~/tmp
cd ~/tmp/
url=https://developer.download.nvidia.com
url+=/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_8.0.61-
1_amd64.deb
wget $url
sudo dpkg -i cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
sudo apt-get update
sudo apt-get install -y --force-yes cuda=8.0.61-1
echo "/usr/local/cuda-8.0/lib64" > sudo /etc/ld.so.conf.d/cuda.conf
sudo ldconfig
```

Installing cuDNN

To install cuDNN:

1. From the NVIDIA cuDNN web page (https://developer.nvidia.com/cudnn), click **Download cuDNN**.
2. Search for "cuDNN v7.0.5 (Dec 5, 2017), for CUDA 8.0" and download the cuDNN v7.0.5 Library for Linux.

   You will download the cudnn-8.0-linux-x64-v7.tgz file.
3. Use the following command to upload files to AWS instance:

   ```
   scp -i aws-test.pem ./cudnn-8.0-linux-x64-v7.tgz ubuntu@<ip-address>:~
   ```
4. Log in to the AWS instance, and use the following commands to install cuDNN 7.05:

   ```
   cd ~
   tar -xzvf cudnn-8.0-linux-x64-v7.tgz
   sudo cp cuda/include/cudnn.h /usr/local/cuda/include
   sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
   sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
   sudo ldconfig
   ```

Installing the Cross Compilation Toolchain

To install the cross compilation toolchain, log in to the AWS instance, and execute the following command:

```
sudo apt-get install -y --force-yes g++-aarch64-linux-gnu
```

Installing the DNNDK Package

To install the DNNDK package, log in to the AWS instance, and execute the following commands:

```
cd ~/xilinx_dnndk_v2.08/host_x86
sudo ./install.sh ZCU102
```

Preparation

Before you begin, it is best practice to use the following commands to back up your folders:

```
cd ~
cp -r xilinx_dnndk_v2.08/ xilinx_dnndk_v2.08_backup
```

Uploading Data

Before you run the resnet50 and inception_v1 examples with DECENT, you must provide calibration dataset. Use the following command to upload data:

```
scp -i aws-test.pem data.tar.gz ubuntu@<ip-address>:~
```

Running DECENT

You can run DECENT commands such as the following (this example is for resnet50):

```
cd ~/xilinx_dnndk_v2.08/host_x86/models/resnet50
tar -xzvf ~/data.tar.gz
./decent.sh
```

Running DNNC

You can run DNNC with the following command:

```
./dnnc.sh
```

Cross Compiling Samples

In the `dnnc_output` folder, compile samples with the following commands:

```
cd ~/xilinx_dnndk_v2.08/ZCU102/samples/resnet50
cp ~/xilinx_dnndk_v2.08/host_x86/models/resnet50/dnnc_output/* ./model/
make
```

Downloading Data from an AWS Instance

When resnet50 `make` is done, you can get its executable binary file. Copy it to local using the following commands:

```
url=/home/ubuntu/xilinx_dnndk_v2.08/ZCU102/samples/resnet50/resnet50
scp -i aws-test.pem ubuntu@<ip-address>:$url ./
```

Copying Samples to the Evaluation Board

Copy the executable binary file to the evaluation board with the following command:

```
scp ./resnet50 root@<ip-address>:~/ZCU102/samples/resnet50
```

Finally, you can log into the evaluation board and see the effect with the following commands:

```
cd ~/ZCU102/samples/resnet50
./resnet50
```

Please Read: Important Legal Notices

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx’s limited warranty, please refer to Xilinx’s Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx’s Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS “XA” IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE (“SAFETY APPLICATION”) UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD (“SAFETY DESIGN”). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.