Amazon AWS FPGA Tutorial

Building for **Cornell ECE 5775: High-Level Digital Design Automation** taught by Professor **Zhiru Zhang**.

Contents

0. Preamble

To meet the increasing requirement of computing ability for heavy computational applications (especially for deep learning realated designs), cloud computing becomes popular in recent years, which enables ubiquitous access to various sharable hardware resources including CPU, GPU, and FPGA. In specific, Amazon Web Services (AWS) is a widely used cloud computing service which offers more than 90 services range among storage, computing, deployment, and etc. This tutorial will introduce how to launch AWS FPGA instances and run designs on it.

Before getting started, we would like to appreciate Joseph Featherston, who created the original version of this tutorial, and Kyle Lu who further improve it based on his own practical experience. We also thank all the valuable advice given by the researchers who were involved when updating the lab assignments of ECE5775.

Enjoy your journey!

1. AWS Account Setup

1.1 Creating a New AWS Account

- Create a new **AWS account** using this link
- You should use your @edu email address to get AWS Educate with $100 free credits
- You will need to add a credit card when creating an account

1.2 Apply for AWS Educate with $100 Free Credits

- Go to this page to apply for **AWS Educate for student**
- Do not choose "AWS Educate Starter Account", you need a regular account instead of a "Starter Account" for full access to the resources
- After filling out all the forms, you will receive an email with a promotion code of **$100 credits**, this usually takes about 10 minutes
- Apply the promotion code following the steps in the email

1.3 Setting up Billing Alarm

- The **AWS Console** is where you can manage basically everything of your AWS account, use this link to go to your AWS Console

- Navigate through `Services -> Billing -> Preferences`, and then click on "Receive Billing Alerts"
- At this Point, you should navigate through the AWS Console to get yourself familiar with its layout and setting

1.4 Plan Ahead: Increase the Instance limit

- We will need an m4 instance and an F1 instance for this course, however our account type only allows us to use 1 instance. So we have to increase this limit before we can proceed
- Use this link to reach out to the customer service for raising the limit
- Fill out the form and wait for AWS staff to process your request
- Please be aware that this could take **2~3 days** for the request to be processed. So, **plan ahead** for your time

2. Launching Development Instance

2.1 Launching New Instance

2.1.1 Find the Launch Entry

- Use this link to open AWS Console
- Navigate through `Services -> EC2`
- Check your zone on the top right, ensure you are in `US East (N. Virginia)`
- Click on "**Launch Instance**"

2.1.2 Choose Amazon Machine Image (AMI)

- Click on "**AWS Marketplace**" on the left, and enter "**FPGA**" in the search bar
- Please use the AMI with Version 1.5.0, which installs the SDAccel 2018.2.
- Select `FPGA Developer AMI` to start configuration
- Scan through the pop-up window and hit "**continue**"

2.1.3 Choose Instance Type

- The lowest recommended instance is `t2.xlarge` (4CPU/16GB ram)
- If you want to run multiple builds in parallel, you probably want one with more RAM, like `m4.2xlarge/m4.4xlarge`, in the course, we recommend to use `t2.2xlarge`
- **A word on why choosing this instance**: In fact, only the f1 series instances are actually connected to a real FPGA, however f1 instances are expensive, given we only have $100 credits so far. Therefore, we start with non-f1 instances and continue with f1 only when we finished all the pre-work
- Hit "**Next: Configure Instance Details**", no need to change anything on this page
- Hit "**Next: add Storage**"

2.1.4 Add Storage

- There are two volumes shown on this page
- **The First Volume** is the 70GB root drive
  - This is where OS and Vivado is installed. Do not modify this volume or save things here. This volume will be deleted once the instance is terminated
- **The Second Volume** is where you will store your work
  - The size should be increased to 30 GB
  - For your information, EBS refers tp "Elastic Blocking Store", you do not need to understand this though
- Hit "**Review and Launch**", since you don't need to change anything in the sections afterwards.
- Review the summary and hit "**Launch**" (It is normal that there is a warning message saying "Your instance configuration is not eligible for the free usage tier")

2.1.5 Create Key Pair

- There will be a pop-up window
- If this is your first time,
  - Select "**Create a new key pair**"
  - Name it something permanent, you will need to reuse it afterwards

    - Carefully download and store it in a safe place on your laptop(or PC), for example `.ssh/`
    - Put down your `path to the key pair` somewhere handy for later use
    - Never let anyone else access this file
- If this is not your first time,
    - you may want to use your existing key pair
- Hit "**Launch Instance**"

2.1.6 Check Your Instance

- Go back to your EC2 Console (Services/EC2)
- Click on "**Instances**" on the left, and you will see you instance showing up there
- Select it and put down the `public DNS` aside with the your key pair path for later use

2.2 Creating AWS Credentials

- Go back to your AWS Console
- Click on your name on the top right, then click on"**My Securtiy Credentials**"
- In the pop-up window, hit "**Get Started with IAM users**"
- Hit "**Add user**", enter a name and select "**Programmatic Access**"
- Add `AmazonEC2FullAccess` and `AmazonS3FullAccess` Permissions
- Copy the `Secret Access key` and `Access Key ID` to a secure location,
    - This is the only time you will be able to access the Secure Access Key if you lose it yßou will need to create a new user
    - These keys will let someone create instances and cause things to be billed to your account so be somewhat careful with them

2.3 Create S3 Bucket

The S3 bucket will be used later by the AWS SDAccel scripts to upload your DCP to AWS for AFI generation which will be packaged into a tar file.

- Open terminal on your laptop (or something alike on PC), ssh into your instance using the following command

```
$ ssh -i <path-to-key-pair> centos@<public dns address>
```

- Install `AWS Command Line Interface (CLI)`, please follow this tutorial
- Configure AWS

```
$ aws configure
```

- When prompted, enter as follows:

```
AWS Access Key ID [None]: <Access Key ID from last step>
AWS Secret Access Key [None]: <Secure Access Key from last step>
Default region name [None]: us-east-1
Default output format [None]: json
```

- Create bucket and relevant folders, the bucket name should be different from any other bucket.

```
# Create an S3 bucket
$ aws s3 mb s3://<bucket-name> --region us-east-1
# Create folder for your tar files
$ aws s3 mb s3://<bucket-name>/<dcp-folder-name>
# Create a temp file
$ touch FILES_GO_HERE.txt
# Put the file in the S3 folder
$ aws s3 cp FILES_GO_HERE.txt s3://<bucket-name>/<dcp-folder-name>/

# Create a folder to keep your logs
$ aws s3 mb s3://<bucket-name>/<logs-folder-name>
# Create a temp file
$ touch LOGS_FILES_GO_HERE.txt
# Put the file in the S3 folder
$ aws s3 cp LOGS_FILES_GO_HERE.txt s3://<bucket-name>/<logs-folder-name>/
```

2.4 Initial Setup

- Check the "**Instance status**" on your EC2 console, proceed to next step when it finishes initialization
  - This takes several minutes from the time it is launched
- Open terminal on your laptop (or something alike on PC), ssh into your instance using the following command

```
$ ssh -i <path-to-key-pair> centos@<public dns address>
```

- The 30GB volume is mounted here `~/src/project_data`

```
$ cd ~/src/project_data
```

- Clone `aws-fpga tools`

```
$ git clone https://github.com/aws/aws-fpga.git
```

- Copy `lab_aws` and `harness` here
- You can use either `scp` or `git clone` to do this.
- Run the `SDACCEL setup script`, this must be done from the aws-fpga root directory

```
$ cd aws-fpga
$ source sdaccel_setup.sh
$ source $XILINX_SDX/settings64.sh
```

2.5 Run Hardware Synthesis

- Go to your working directory and find the `typedefs.h` file in src, make sure `TARGET_DEVICE` is `xilinx_aws-vu9p-f1-04261818_dynamic_5_0`

```
$ cd ~/src/project_data/lab_aws/
$ cd src/host
$ vi typedefs.h
```

- Run the hardware synthesis
  - This takes about **4 hours**
  - Do not disconnect from the instance while doing this

```
$ make ocl OCL_TARGET=hw OCL_PLATFORM=$AWS_PLATFORM APPLICATION_DIR=.
```

- If you run into problem while doing this, and you are using a fairly new laptop, try run the following commands, and then restart from Section 2.2

```
$ export LC_CTYPE=en_US.UTF-8
$ export LC_ALL=en_US.UTF-8
```

- You can add these lines to `/etc/environment` to fix this problem permanently

```
  LANG=en_US.utf-8
  LC_ALL=en_US.utf-8
  LC_CTYPE=en_US.UTF-8
```

2.6 Generating Amazon FPGA Image (AFI)

- Run `Configure AWS` again if you started a new terminal session

```
$ aws configure
AWS Access Key ID [None]: <Access Key ID from last step>
AWS Secret Access Key [None]: <Secure Access Key from last step>
Default region name [None]: us-east-1
Default output format [None]: json
```

- Submit the job to create FPGA image by running

```
$ $SDACCEL_DIR/tools/create_sdaccel_afi.sh \
-xclbin= <input_xilinx_fpga_binary_xclbin_filename> \
-o=<output_aws_fpga_binary_awsxclbin_filename_root> \
-s3_bucket=<bucket-name> \
-s3_dcp_key=<dcp-folder-name> \
-s3_logs_key=<logs-folder-name>
```

- You can check the status of the run by running the following command, it lists all your fpga images

```
$ aws ec2 describe-fpga-images --owners self
```

- Alternatively, you can look up specific pga image by its `<AFI ID>`, which is shown in `<timestamp>_afi_id.txt` created when script runs

```
$ aws ec2 describe-fpga-images --fpga-image-ids <AFI ID>
```

- You need to wait for “**State**” to change from "**pending**" into “**available**” before you can use the image
  - This could take about **1 hour**

2.7 Stopping the Instance

- Go back to AWS Console and navigate to the instances page
- Stopping or termination instances
  - You can stop or terminate the instance by right click on the instance and select corresponding action
  - You can resume stopped instances while terminated instances are gone
- Charges
  - You are not charged for an instance in stopped or terminated state
  - You are charged for instances in the running state, volumes and snapshots(i.e. backups) so you should check that there isn’t anything there you don’t need
  - You do need to pay for the 70GB whenever the instance is in the stopped state The price of EBS is ~$0.10/GB-month, so the volume will cost $7/month, while it is probably worth spending the $3/month to keep your work volume around if you are using it relatively frequently

3. Running on Hardware

3.1 Launch F1 Instance

- The process is the same with `Section 2.1`, except the following:
- At "Step 2: Choose an Instance Type": choose `f1.2xlarge` instead of previous instance type
  - These instances cost $1.65/hour so try to limit how long you have them running
- At "Step 3: Configure Instance Details": in the subset dropdown, make sure you select the same availability zone as the previous 30GB volume
  - You can check the availability zone of the previous 30GB volume from `EC2 Console -> Volume`
- At "Step 4: Add Storage": we are going to use the previous 30GB volume we have created, therefore we don't need the second 5GB volume that added by default. So remove the second volume.

3.2 Mount Previous Volume

- Navigate to to volume page `EC2 Console -> Volume`
- Right click your 30GB volume, click on "**Detach**"
- Go back to your instances and figure out the id of you f1.2 instance (you don't need to memorize the id, you only need to be able to tell which id is of your f1.2 instance, which one is of you m4.2 instance)
- Right click your 30GB volume, click on "**Attach**", then select the id of your f1 instance
- Open terminal on your laptop (or something alike on PC), ssh into your f1 instance using the following command with correct `public dns address`

  ```
  $ ssh -i <path-to-key-pair> centos@<public dns address>
  ```

- Run the following command to see the block devices, find the one which is 30GB, put down the `path of your block`, which in my case is `/dev/xvdf`

  ```
  $ lsblk
  ```

- We would like to mount our previous 30GB volume to the same directory with before for consistency, which is `~/src/project_data`, however the AWS f1 instance won't let us do it for whatever reason it is. So we are using the following steps to work it around
  - Go to `~/src/` and create a new directory named `project`

```
$ cd ~/src
$ mkdir project
```

- Mount the 30GB volume to `~/src/project`

  ```
  $ sudo mount /dev/xvdf ~/src/project
  ```

- Copy the fold `aws-fpga` from `~/src/project` to `~/src/project_data`

  ```
  $ cp -r aws-fpga/ ../project_data/
  ```

- Note that we only need fold `aws-fpga` to be in directory `~/src/project_data`, because some inner codes of `aws-fpga` are hard-coded and bonded up to this directory (not our fault), we can surely change the inner code, however it is to much work to do so, therefore we are doing this as a workaround.

3.3 Run Instance

- Run SDAccel Setup if you haven't since rebooting

  ```
  $ source ~/src/project_data/aws-fpga/sdaccel_setup.sh
  ```

- Switch to root shell

  ```
  $ sudo sh
  $ source /opt/xilinx/xrt/setup.sh
  ```

- Go to your working directory and run the application following the instruction in `README.md`, which is shown below in our case lab_aws

  ```
  $ cd /home/centos/src/project/lab_aws
  $ ./cordic_host.exe -f <path_to_xclbin_file>
  ```