Xilinx 2D RTM Application Userguide

Liang Ma[1], **Yifei Zhou**[2], **and Lisa Liu**[3]

[1, 3]DCG, Xilinx, Ireland
[2]DCG, Xilinx, USA

1 Introduction

Reverse Time Migration (RTM) is an important seismic imaging technique used for producing an accurate representation of the subsurface. We adopted the methodology by solving **acoustic wave equation in time-domain with hybrid boundary condition** for the migration. We designed this APP for users to evaluate the performance of Xilinx FPGA Alveo U280 in RTM applications.

2 Application Usage

The application is containerized and can be easily run in a few minutes in the cloud or on premises. Details can be found at link https://www.xilinx.com/products/acceleration-solutions/2d-reverse-time-migration.html.

2.1 Sample

The command to run RTM on sample datasets is

```
$ /opt/xilinx/rtm2d/run.sh
```

The user needs to read and accept the license printed in the terminal in order to continue the appliation.

The sample datasets located in */opt/xilinx/rtm2d/dataset_sample* are based on Pluto model which is shown in Figure 1 with dimension data listed in the Table 1. The extended dimension is 1280x7040.

Model name	Height	Width	Time steps	No. shots	Boundary depth	Estimate execution time [s]
Pluto	1200	6960	12860	5	40	220

Table 1. Sample model dimensions

2.2 Prerequisites

The bash script calls the python script *rtm_kernel.py*, which parses the configuration from the input file *config.json*. According to the configuration, the python script calls *host.exe*, which reads data from a given path and starts the kernel *rtm.xclbin*, as shown in Figure 2.

2.2.1 Configurations

An example configuration file is */opt/xilinx/rtm2d/config.json*. The application use the configurations in the **JSON** file for migration. Table 2 lists the main entries in the configuration file. The extended pressure filed dimensions are given by $depth = nz + 2 * nzb$ and $width = nx + 2 * nxb$.

2.2.2 Datasets

If the entry **random** in the configuration file is set **true**, the application will use random data for the migration. With random datasets, one can easily evaluate the RTM app execution time for given configurations.

If the users wish to use their own data instead of random datasets for the migration, please prepare all the files listed in the Table 3 and put them in the dataset's path specified in the configuration.

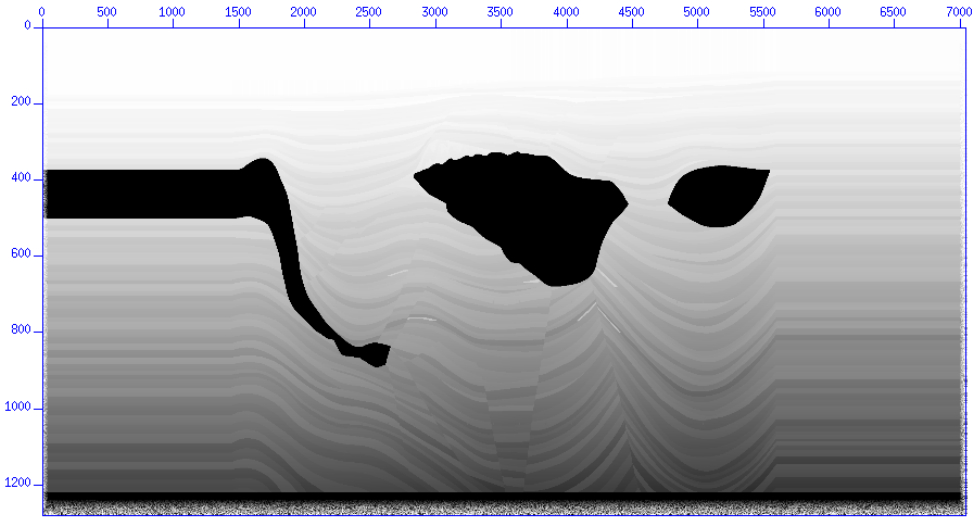

Figure 1. Pluto model (1280x7040 with boundary extension)

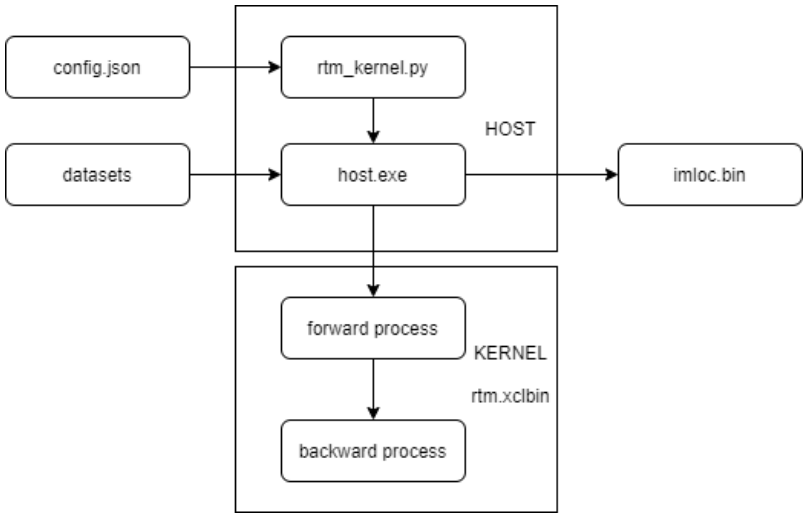

Figure 2. RTM application block diagram

2.2.3 FPGA and Device ID

This application supports Xilinx FPGA Alveo U280 card at this moment. To run this application on users' machines, please make sure:

- Xilinx FPGA Alveo U280 card is installed correctly
- Xilinx Runtime (XRT v2019.2) is installed correctly. The default installation path is */opt/xilinx/xrt/*.

If **Alveo U280** is the only FPGA installed on the machine, the default device id is 0. If not sure, please use the following command to find the device id of 'Alveo U280' card on the machine and set the value in the configuration file.

```
$ source /opt/xilinx/xrt/setup.sh
$ xbutil scan
```

2.3 Run Application

Once the prerequisite configuration file and datasets are prepared, please use the following commands to run RTM application,

```
$ /opt/xilinx/rtm2d/run.sh ./config.json
```

Main Keys	Keys	Notes
model	name	Model name
	nx	Pressure filed width (No. grids)
	nz	Pressure filed depth (No. grids)
	nt	No. time steps
	nshots	No. total shots in Migration
	fsp	First shot position
	sp	Shot position increment
	nxb	Boundary width
	nzb	Boundary depth
	dx	Grid size along width
	dz	Grid size along depth
	dt	Time interval
kernel	deviceId	Device id installed on the machine
dataset	random	Use random data for migration if value is **true**
	path	Path to load and store data

Table 2. Configuration file

File name	Notes
v2dt2.bin	column-major extended processed velocity model, v^2dt^2
ext_vel.bin	column-major extended velocity model, if *v2dt2.bin* is not present
src_s%d.bin	source data for each shot, %d is 0, 1, 2 ...
sensor_s%d.bin	sensor/receiver data for each shot, %d is 0, 1, 2 ...
coefx.bin	stencil coefficients along x multiplied by *dx*, optional
coefz.bin	stencil coefficients along z multiplied by *dz*, optional
taperx.bin	taper coefficients on upper boundary along x, optional
taperz.bin	taper coefficients on upper boundary along z, optional

Table 3. Required files

or

```
$ /opt/xilinx/rtm2d/python/run.py --config ./config.json
```

The user needs to accept the license in order to continue to run the appliation. In order to skip to see the license, please use the command showing below.

```
$ /opt/xilinx/rtm2d/python/run.py --config ./config.json --accept-EULA
```

2.4 Results

Once the application finished, one can checkout the total execution time, average forward process execution time and backward process execution time on the screen. The migration result *imloc.bin* is stored to the dataset's path specified in the configuration file or in the current working directory if program doesn't have write permission to the specified path. Figure 3 is the measured execution time printed in the terminal. Figure 4 and Figure 5 are the migration image of the sample dataset.

```
Accept the license [y/n]:y
*******************************
This 2D RTM model is based on time-domain solution with stencil order 8 and hybrid boandary condition.
The input velocity model dimension (with boundary extension) is: depth 1280, width 7040 and time 12860
No.shots: 5
First shot position: 3520
*******************************
Host and xclbin path: /opt/xilinx/rtm2d/
Data path is: ./dataset_sample/
Starting migration model pluto, estimated total execution time is: 212s.
Execution completed
one forward execution time 19.4303s.
one backward execution time 38.7434s.
Average async execution time 38.4887s.
Total execution time 220.986s.
```

Figure 3. Execution time shown in the terminal

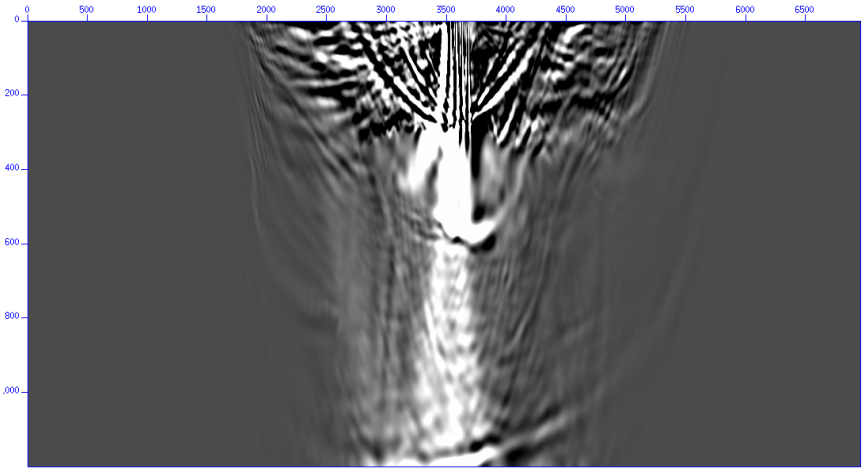

Figure 4. Pluto model migration result image

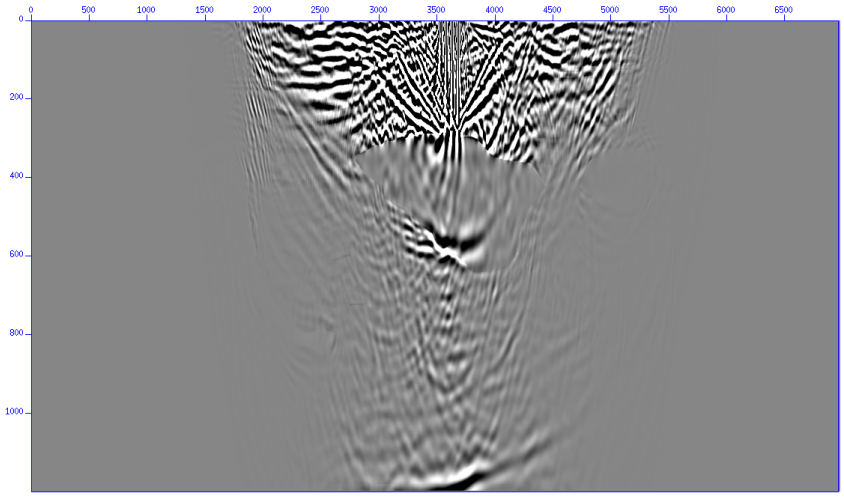

Figure 5. Pluto model migration image (filtered)