Exploring FPGAs in the Open Cloud Testbed: A Hands-On Tutorial

Open Cloud Testbed
Exploring Next-Generation Cloud Platforms

Miriam Leeser

Suranga Handagala

mel@coe.neu.edu

suranga@northeastern.edu

AMD XILINX

Northeastern University

Sign up for an account!

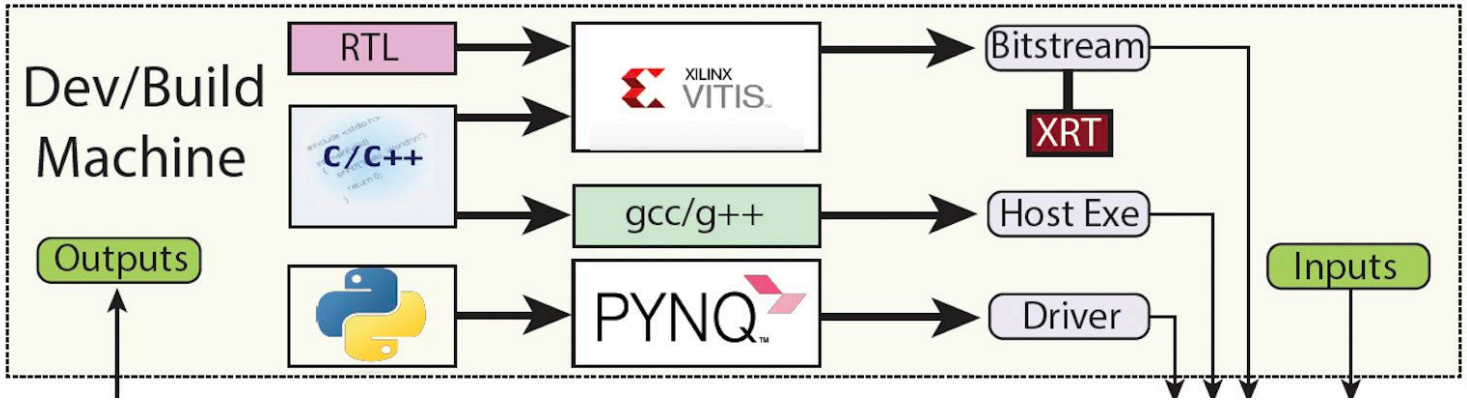
- generate both a public key and a private key.
- Share your **public key** with s.handagala@northeastern.edu.
- Suranga will send you account details and information about the tutorial

Outline

- Introduction to FPGAs in OCT
- Dev/Build workflow
- Targeting workflow
- Examples
  - Vector addition (hello world)
  - Encryption and decryption using two FPGAs
- Q&A

OCT FPGA SW/HW

Build



Deployment

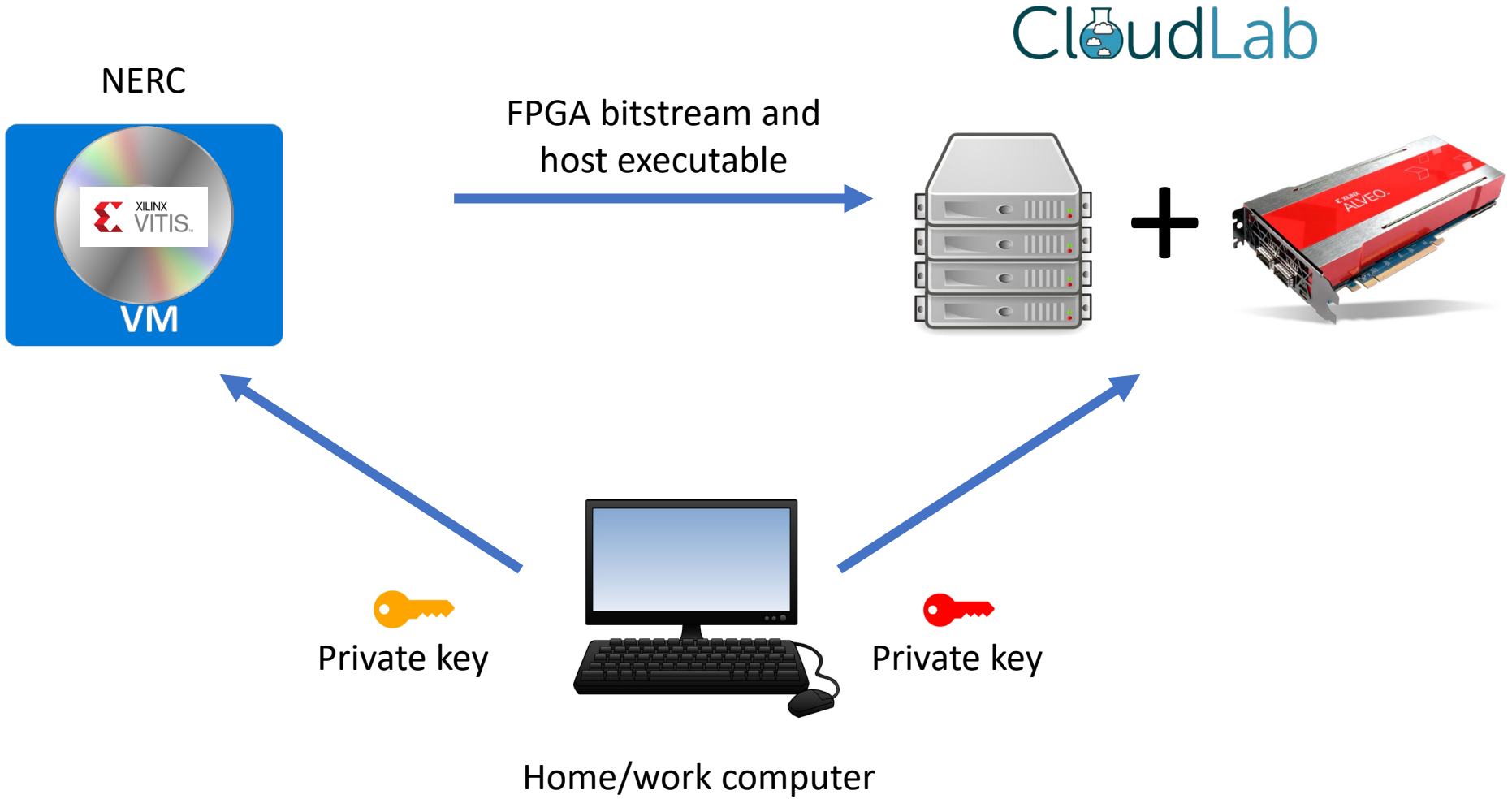
CloudLab

+

Alveo U280
FPGA accelerators

Zink, Michael, David Irwin, Emmanuel Cecchet, Hakan Saplakoglu, Orran Krieger, Martin Herbordt, Michael Daitzman, Peter Desnoyers, Miriam Leeser, and Suranga Handagala. "The Open Cloud Testbed (OCT): A platform for research into new cloud technologies." In *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*, pp. 140-147. IEEE, 2021.

Overall Setup



Start a Vitis Build

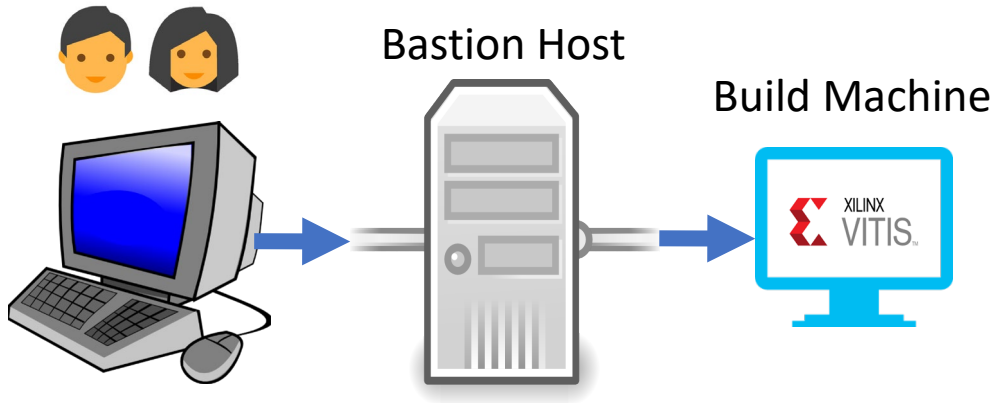
- Build Machines are on NERC: **N**ew **E**ngland **R**esearch **C**loud

- Get a user name and IP address from Suranga
  - You need to generate an SSH key pair and share your public key with us
  - Walk in accounts are also available
- The example we are doing is the Hello world (vector addition) example
  - `git clone https://github.com/Xilinx/Vitis_Accel_Examples`
  - SW emulation, HW emulation, HW execution

Focus of the tutorial

- See the handout for instructions

Build Workflow



- Connect to the build machine

```
ssh -o ProxyCommand="ssh -i <private_key> -W %h:%p
<username>@<bastion_host>" -i <private_key> <username>@<build_machine>
```

- Start VNC server

```
vncserver -geometry 1920x1080
```

← Specify the resolution

- Connect to the VNC server

```
ssh -L <local_port>:localhost:<remote_port> -o ProxyCommand="ssh -i
<private key> -W %h:%p <username>@<bastion_host>" -i <private_key>
<username>@<build_machine>
```

Xilinx Vector Addition Example

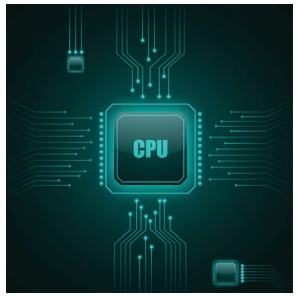
```
make all TARGET=<sw_emu/hw_emu/hw>
PLATFORM=/opt/xilinx/platforms/xilinx_u280_xdma_201920_3
/xilinx_u280_xdma_201920_3.xpfm
```

Build both host executable and bitstream

Hardware build

g++ → Host executable *(runs on host)*

v++ → FPGA bitstream *(runs on Alveo U280)*

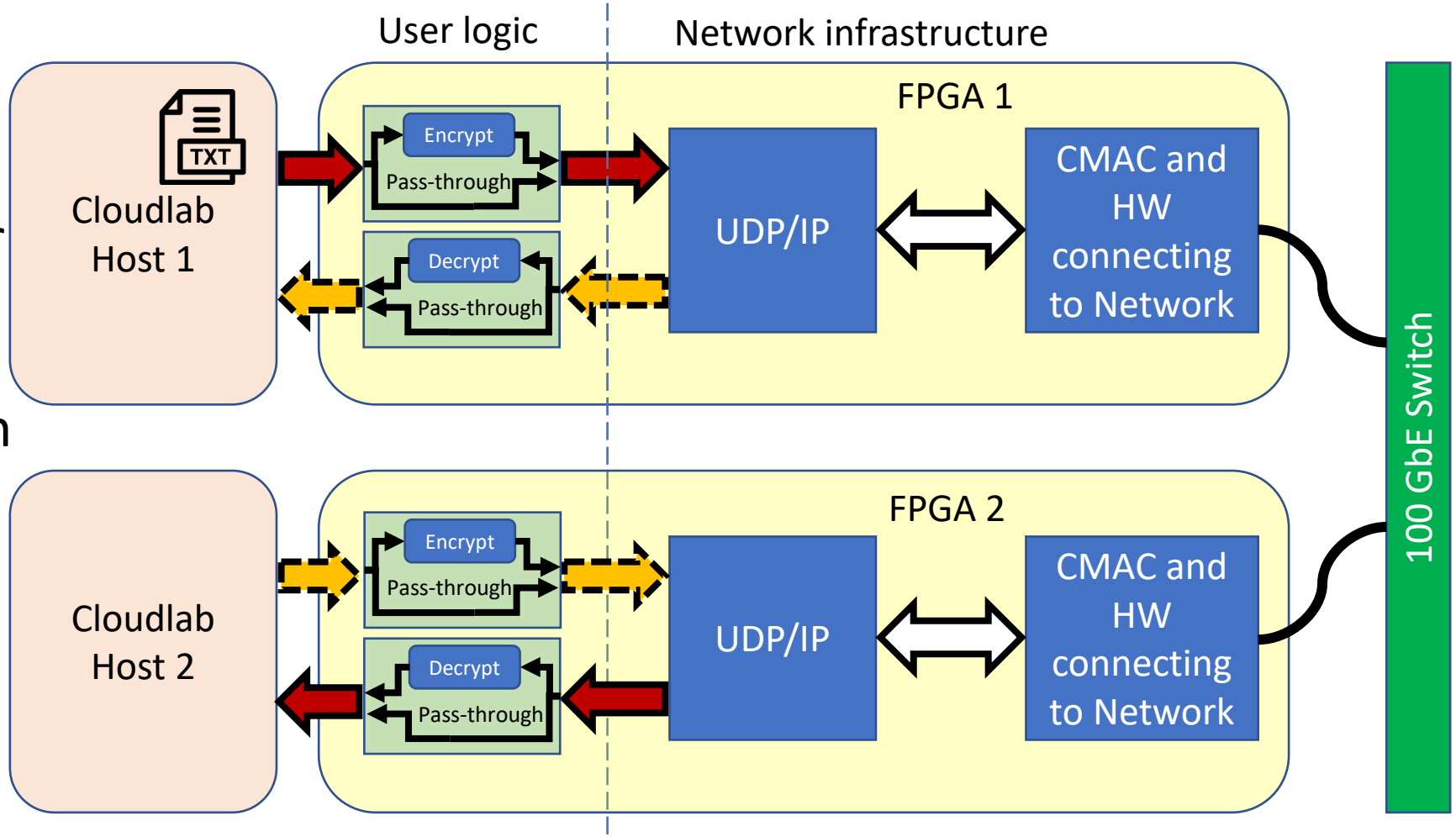


`hello_world`

`vadd.xclbin`

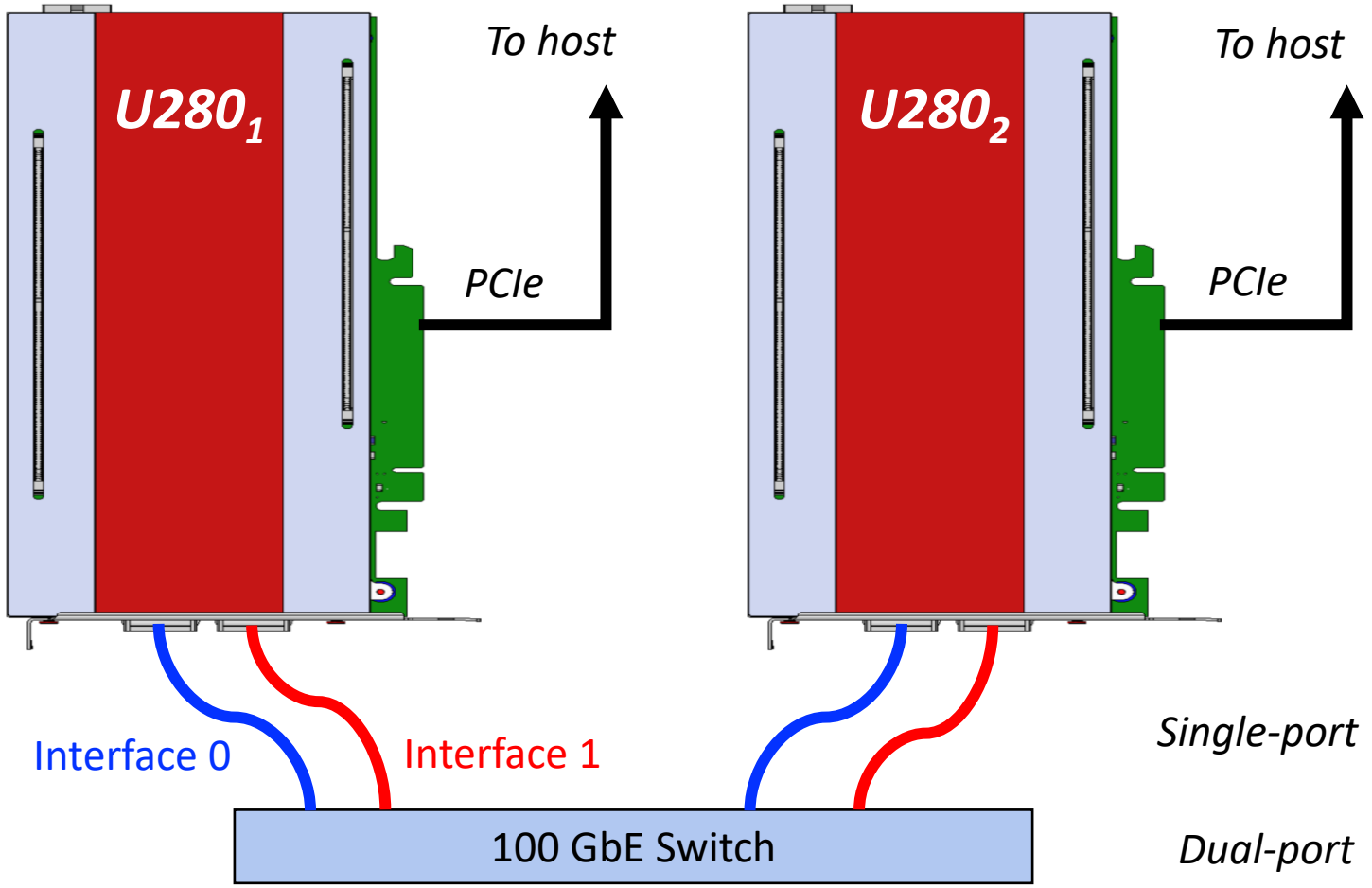
Encryption-Decryption Demo

- Read a text file from host memory
- Encrypt and send over the network
- AES-128 encryption/decryption
- Receiver decrypts the data



UDP Stack: https://github.com/Xilinx/xup_vitis_network_example
Encrypt/decrypt logic: https://github.com/hplp/AES_implementations

UDP Encryption/Decryption Example




```
make all INTERFACE=<0,1,3>
```

g++	v++
Host executables	Bitstream
host_receiver_if0 Host_sender_if0	demo_if0.xclbin

Single-port

```
make all INTERFACE=0
make all INTERFACE=1
```

Dual-port → `make all INTERFACE=3`

Build Both Examples

- Hello world (vector addition) example
  - `git clone https://github.com/Xilinx/Vitis_Accel_Examples`
  - SW emulation, HW emulation, HW execution
- Encryption-decryption demo
  - UDP stack by Xilinx
  - `git clone https://github.com/OCT-FPGA/udp-network-demo`
  - HW execution only

Open Cloud Testbed

Open Cloud Testbed

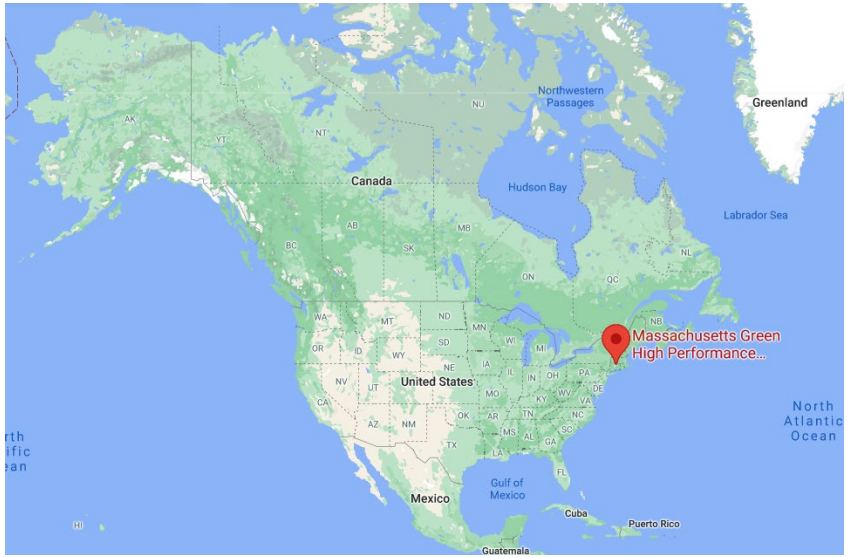
Exploring Next-Generation Cloud Platforms

- Builds on
  - NERC: New England Research Cloud
  - VMs created for users
- Deployment
  - CloudLab
  - Free for US researchers
  - Bare-metal servers
  - Large scale experiments
  - Federate with other testbeds (CloudLab, Fabric, Chameleon Cloud)

NERC

CloudLab

Facility - MGHPCC



NERC

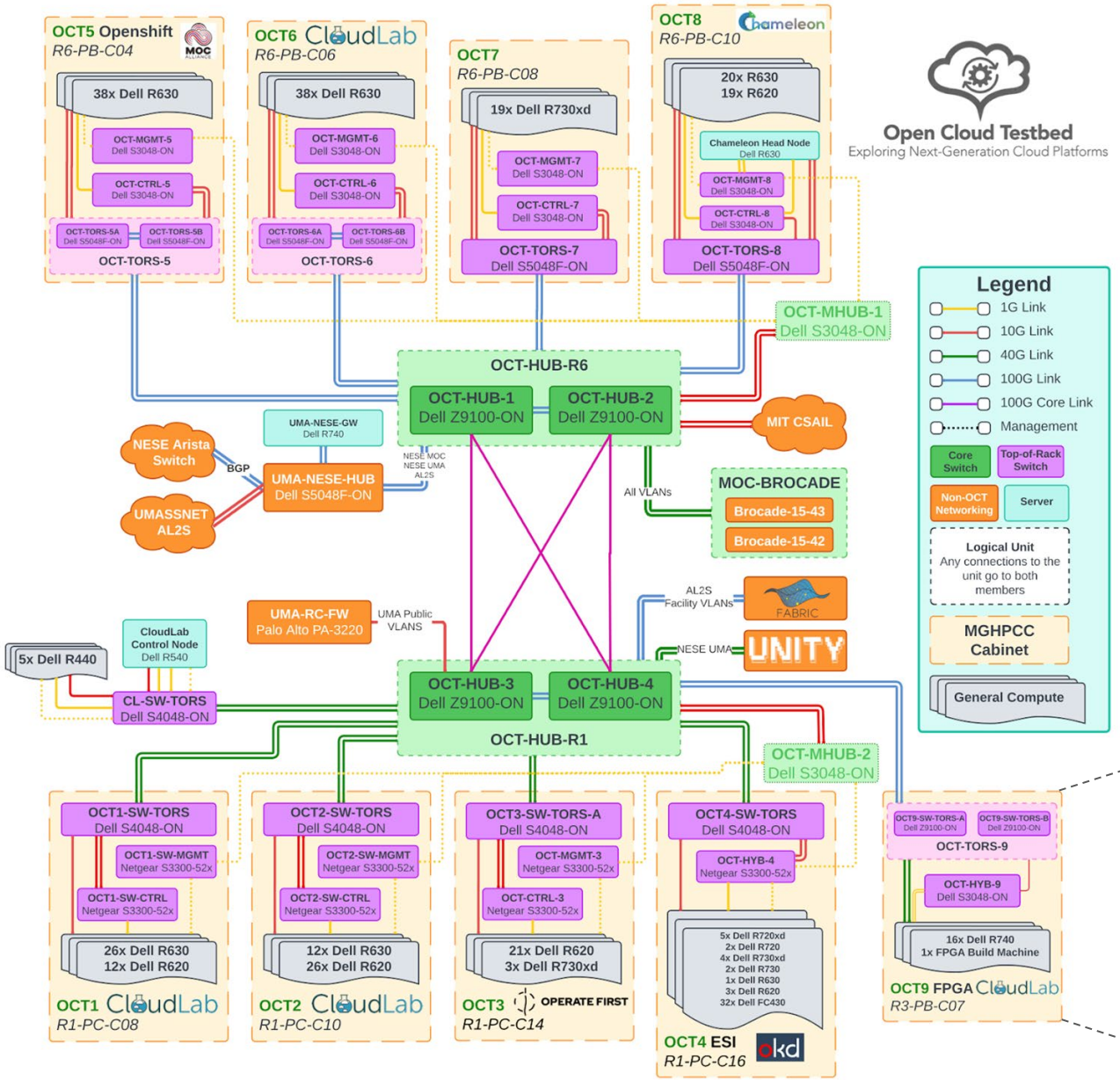
nerc.mghpcc.org

CloudLab

cloudlab.us

- **M**assachusetts **G**reen **H**igh **P**erformance **C**omputing **C**enter – Holyoke MA
- NERC – Persistent storage for hosting tools
- CloudLab – For flexible cloud research (Testbed itself is a part of CloudLab)

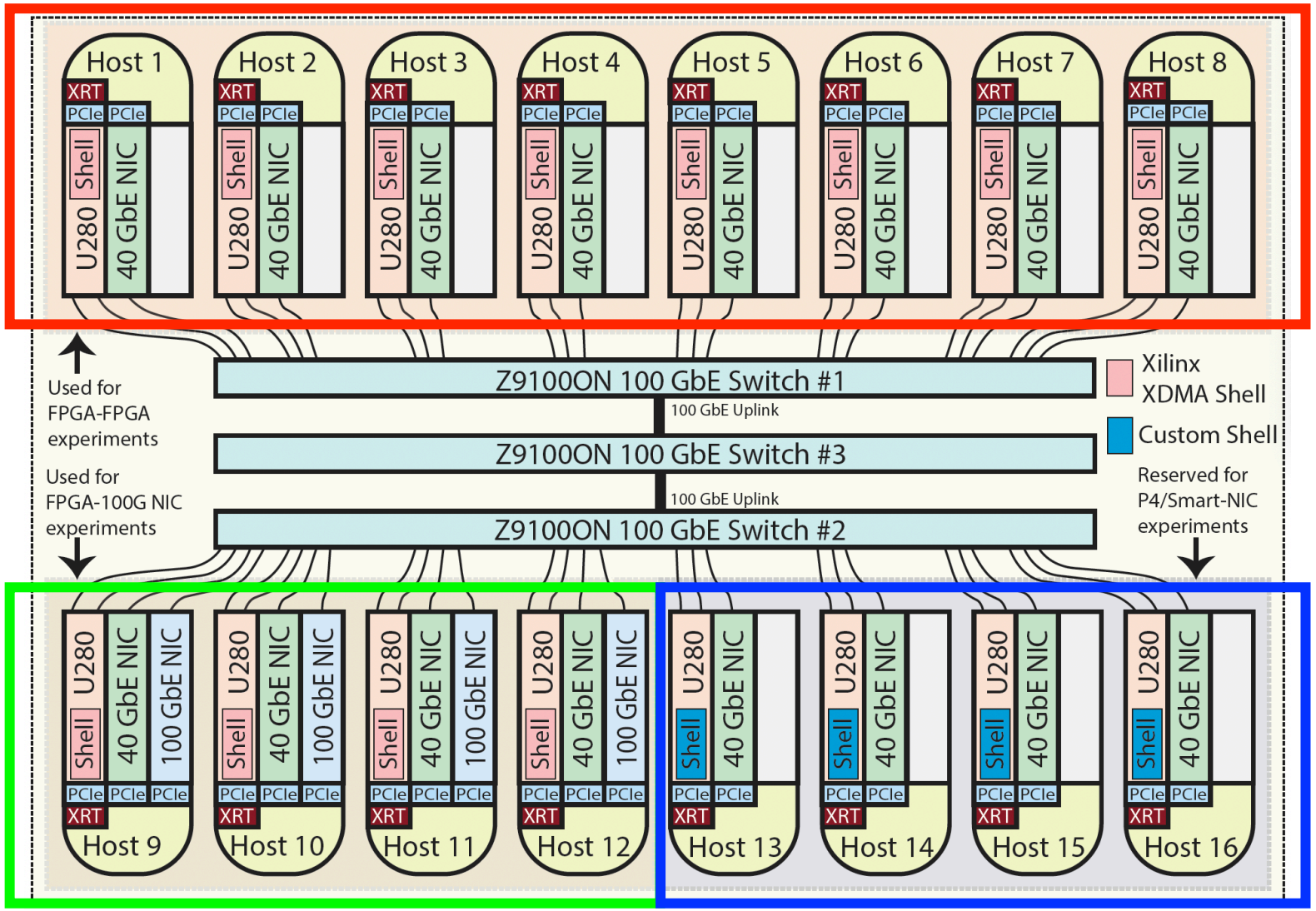
Open Cloud Testbed (OCT)



CloudLab

https://www.cloudlab.us/

Architecture of OCT FPGAs



- Sixteen Nodes
  - Intel Xeon 2.9 GHz, 64 cores
  - 196 GB DDR 4 RAM
- Each node has an Alveo U280
  - PCIe connected
  - Directly connected to the 100 GbE network

U280 + 40 GbE NIC
Xilinx XDMA Shell

U280 + 40 GbE NIC + 100 GbE NIC
Xilinx XDMA Shell

U280 + 40 GbE NIC
Custom Shell – P4/SmartNIC

XILINX VITIS

XILINX VITIS

VIVADO

Build Machine (NERC)

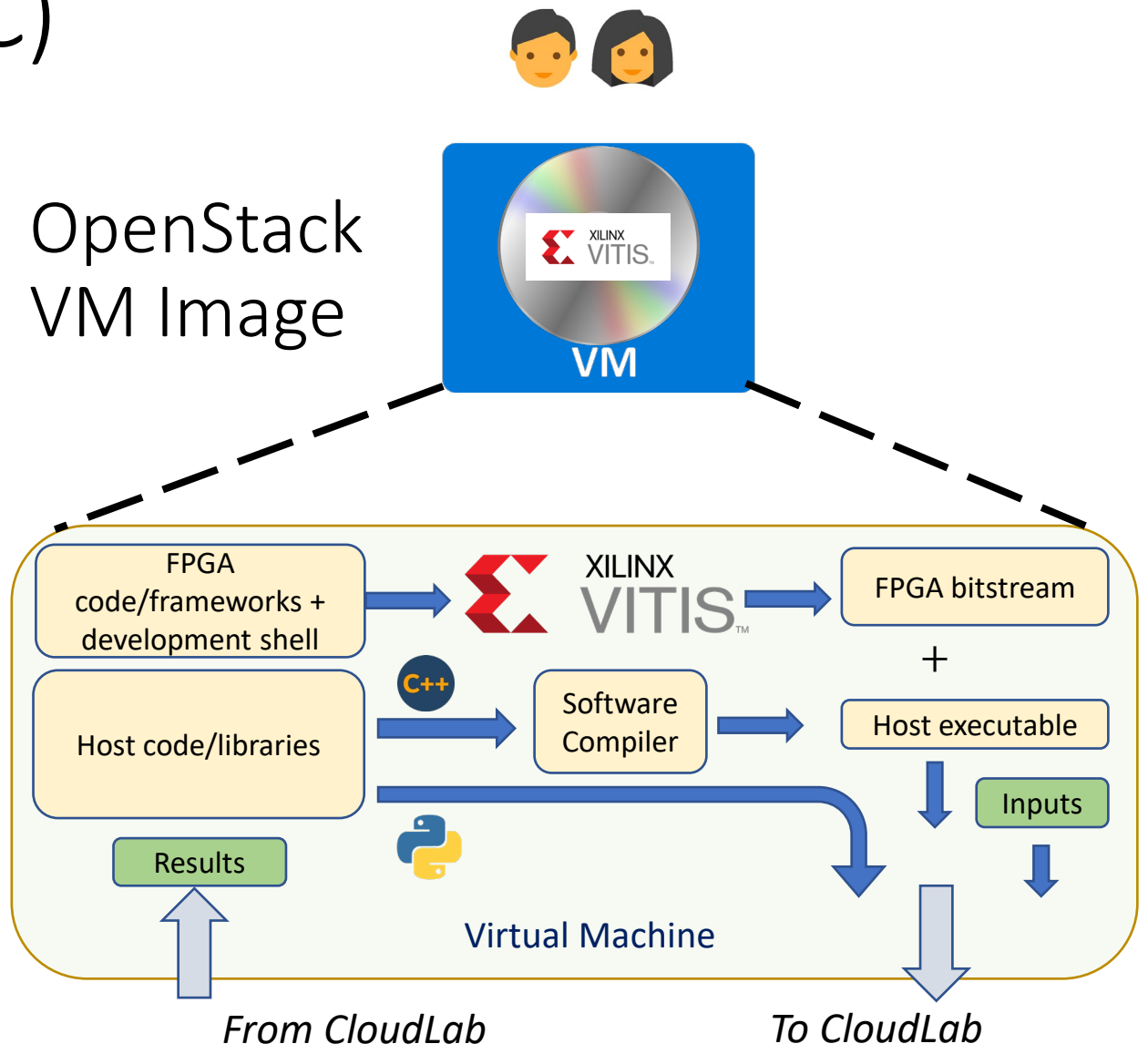
- Xilinx Vitis
  - Vitis
  - Vivado
  - Vitis HLS

Tools

- Xilinx Runtime (XRT)
- Deployment target platform
  - The communication layer required to interact with the card
- Development target platform
  - Used to build custom applications for U280 FPGAs

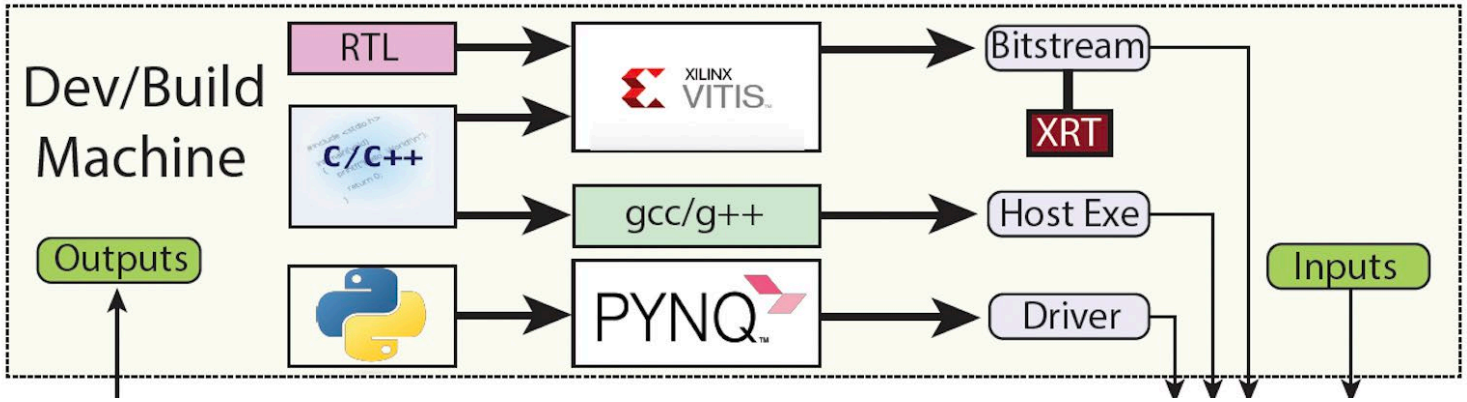
https://www.xilinx.com/products/boards-and-kits/alveo/u280.html#gettingStarted

OpenStack VM Image



OCT FPGA SW/HW

Build



Deployment

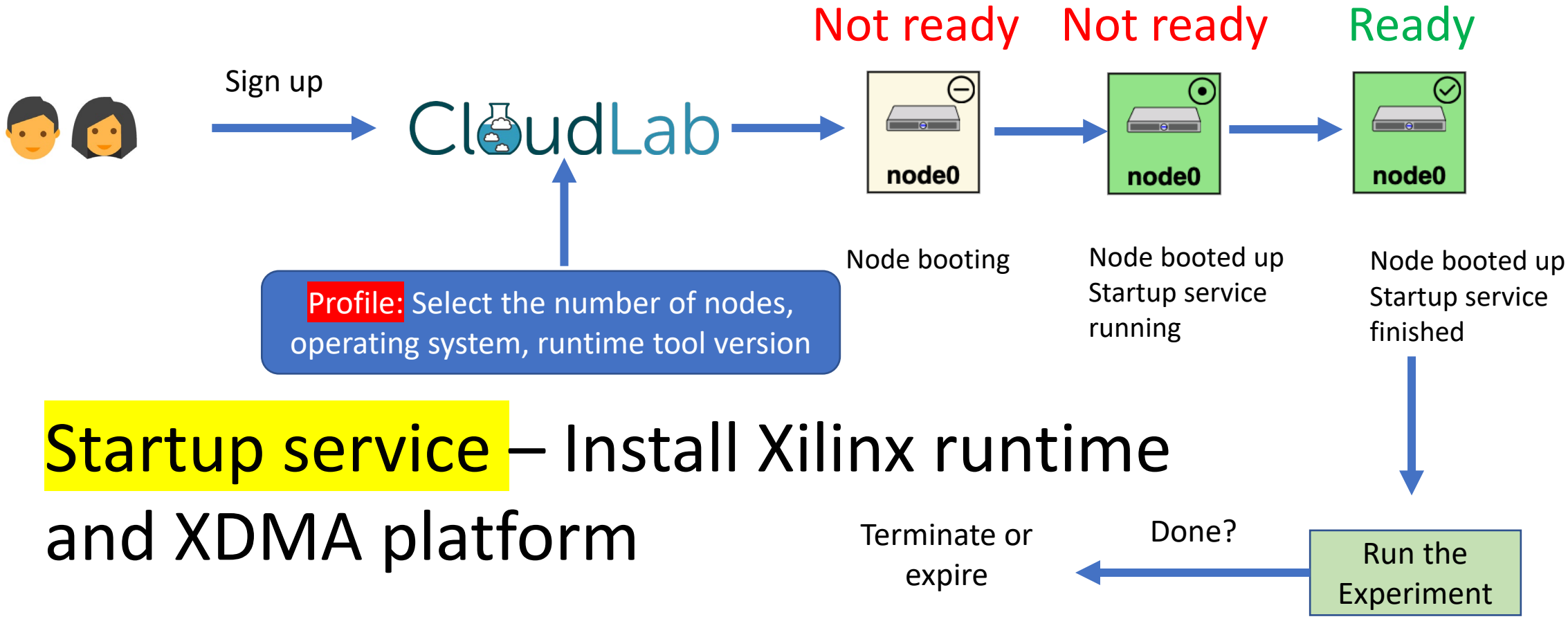
CloudLab

+

Alveo U280
FPGA accelerators

Zink, Michael, David Irwin, Emmanuel Cecchet, Hakan Saplakoglu, Orran Krieger, Martin Herbordt, Michael Daitzman, Peter Desnoyers, Miriam Leeser, and Suranga Handagala. "The Open Cloud Testbed (OCT): A platform for research into new cloud technologies." In *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*, pp. 140-147. IEEE, 2021.

Deployment Workflow



Startup service – Install Xilinx runtime and XDMA platform

CloudLab experiments are time-limited.

Xilinx Vector Addition Example

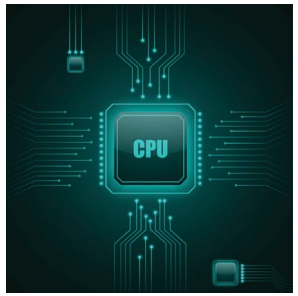
```
make all TARGET=<sw_emu/hw_emu/hw>
PLATFORM=/opt/xilinx/platforms/xilinx_u280_xdma_201920_3
/xilinx_u280_xdma_201920_3.xpfm
```

Build both host executable and bitstream

Hardware build

g++ → Host executable *(runs on host)*

v++ → FPGA bitstream *(runs on Alveo U280)*



`hello_world`

`vadd.xclbin`

Build is Complete … Move to Deployment

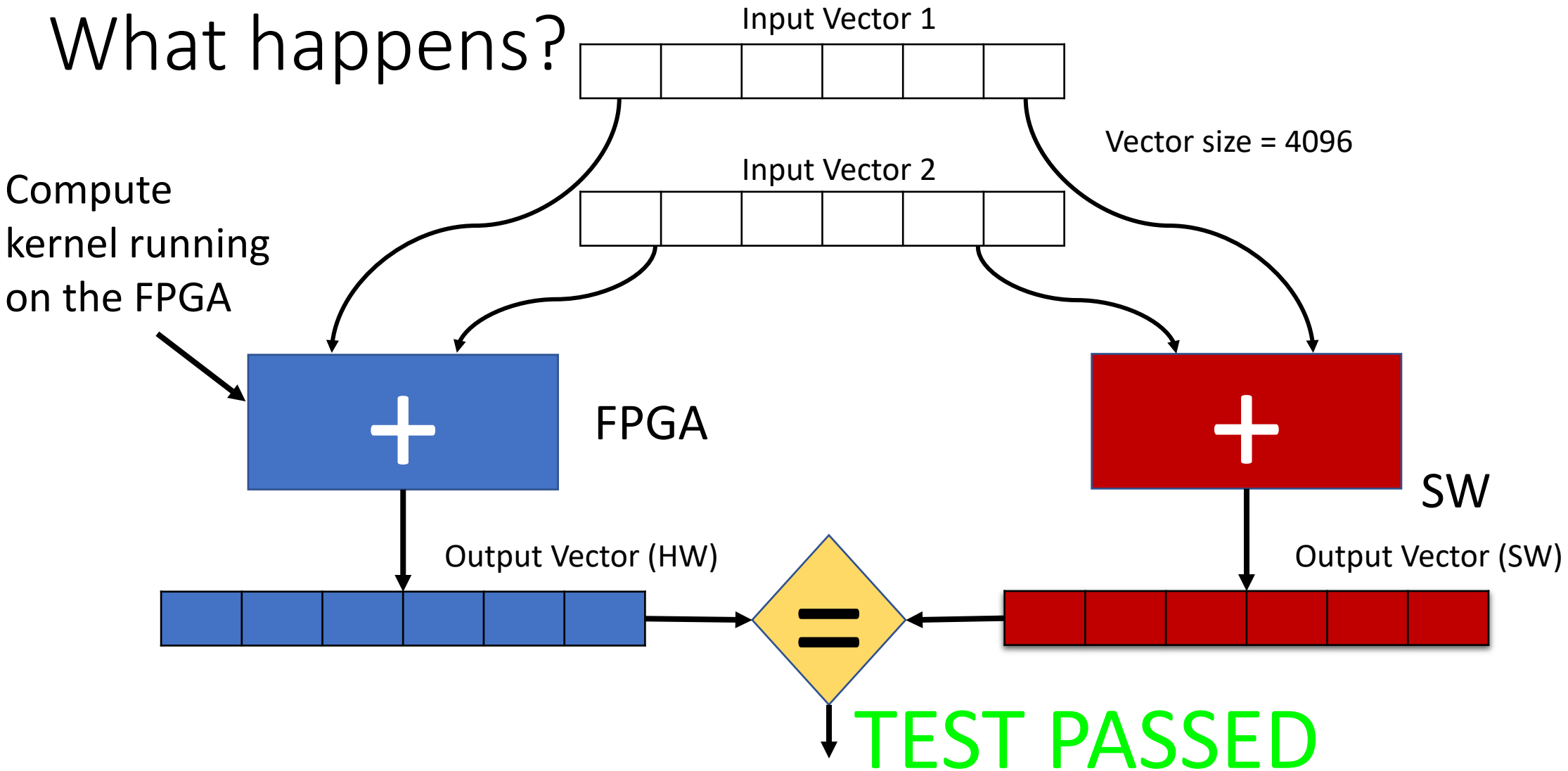
- Move bitstream, host executable to CloudLab

```
scp build_dir.hw.xilinx_u280_xdma_201920_3/vadd.xclbin
hello_world <username>@<CloudLab IP>:
```

- If your build did not complete use prebuilt version

`https://github.com/OCT-FPGA/prebuilt-files`

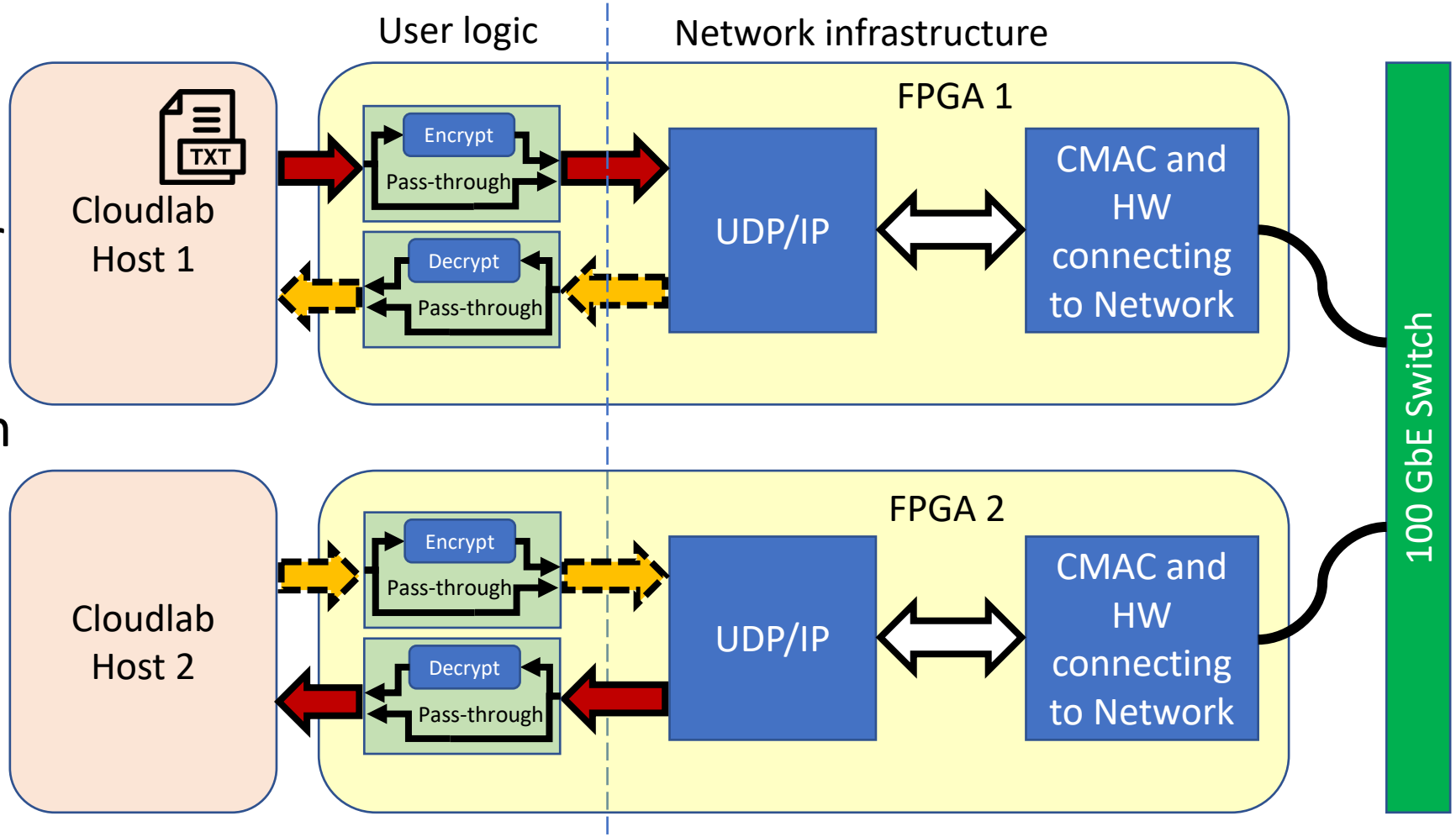
What happens?



Host code: https://github.com/Xilinx/Vitis_Accel_Examples/blob/master/hello_world/src/host.cpp
Compute kernel (FPGA) code: https://github.com/Xilinx/Vitis_Accel_Examples/blob/master/hello_world/src/vadd.cpp

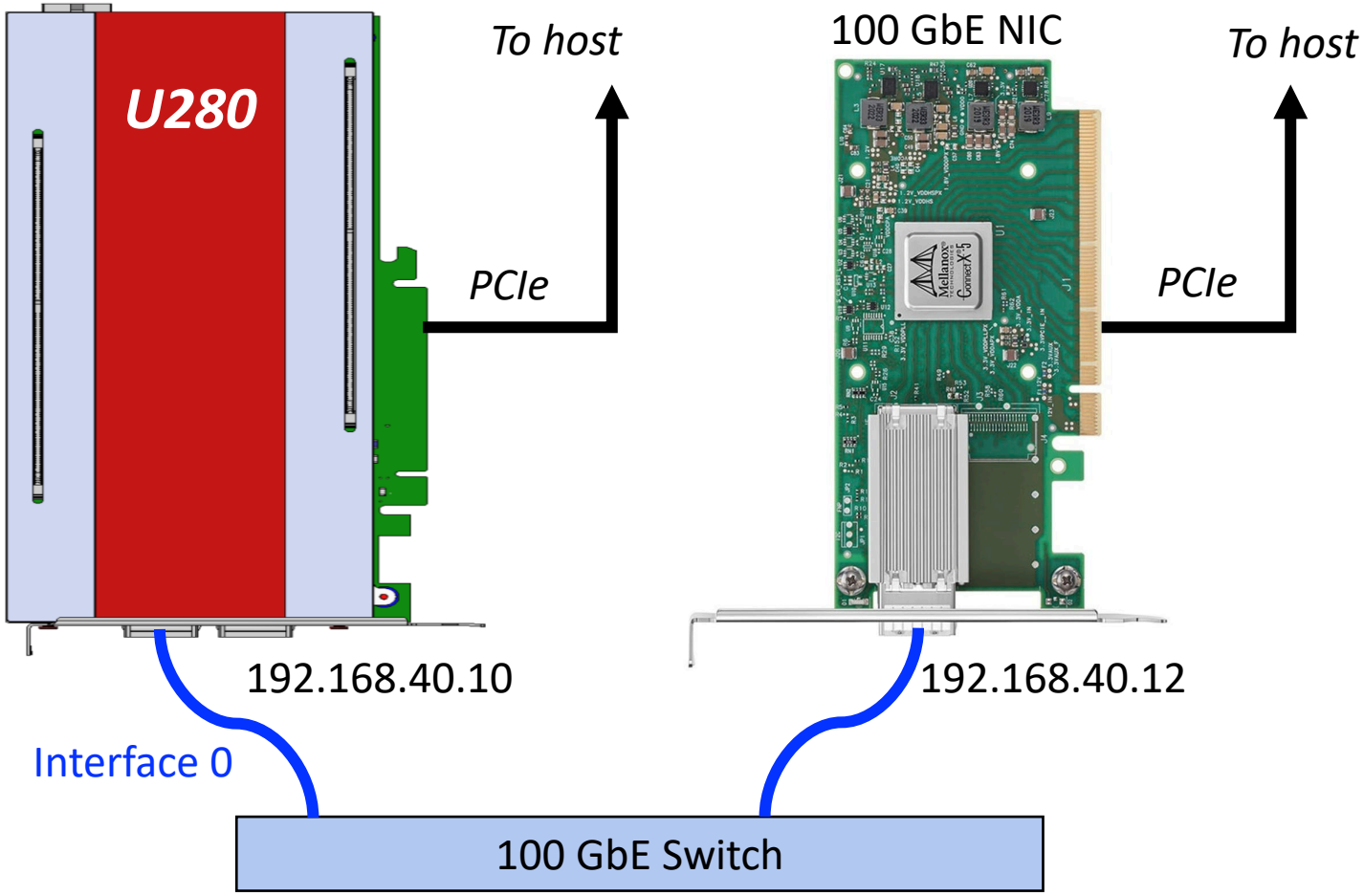
Encryption-Decryption Demo

- Read a text file from host memory
- Encrypt and send over the network
- AES-128 encryption/decryption
- Receiver decrypts the data



UDP Stack: https://github.com/Xilinx/xup_vitis_network_example
Encrypt/decrypt logic: https://github.com/hplp/AES_implementations

VNx Basic Example: FPGA to NIC (and vice versa)



- UDP basic example by Xilinx
- FPGA and NIC should be on the same host.

Details here:

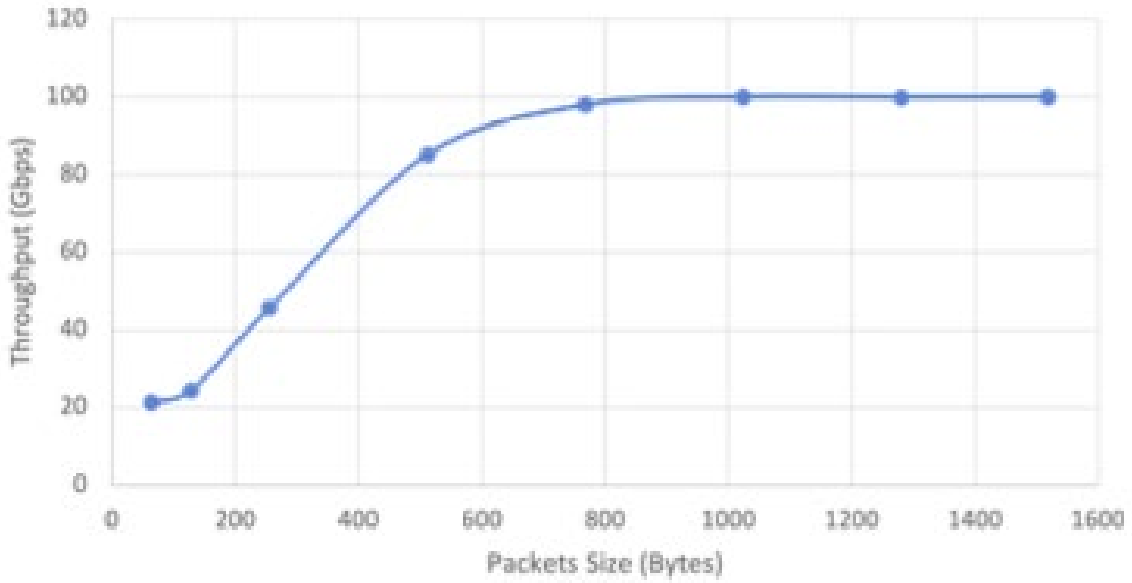
https://github.com/Xilinx/xup_vitis_network_example/blob/master/Notebooks/vnx-basic.ipynb

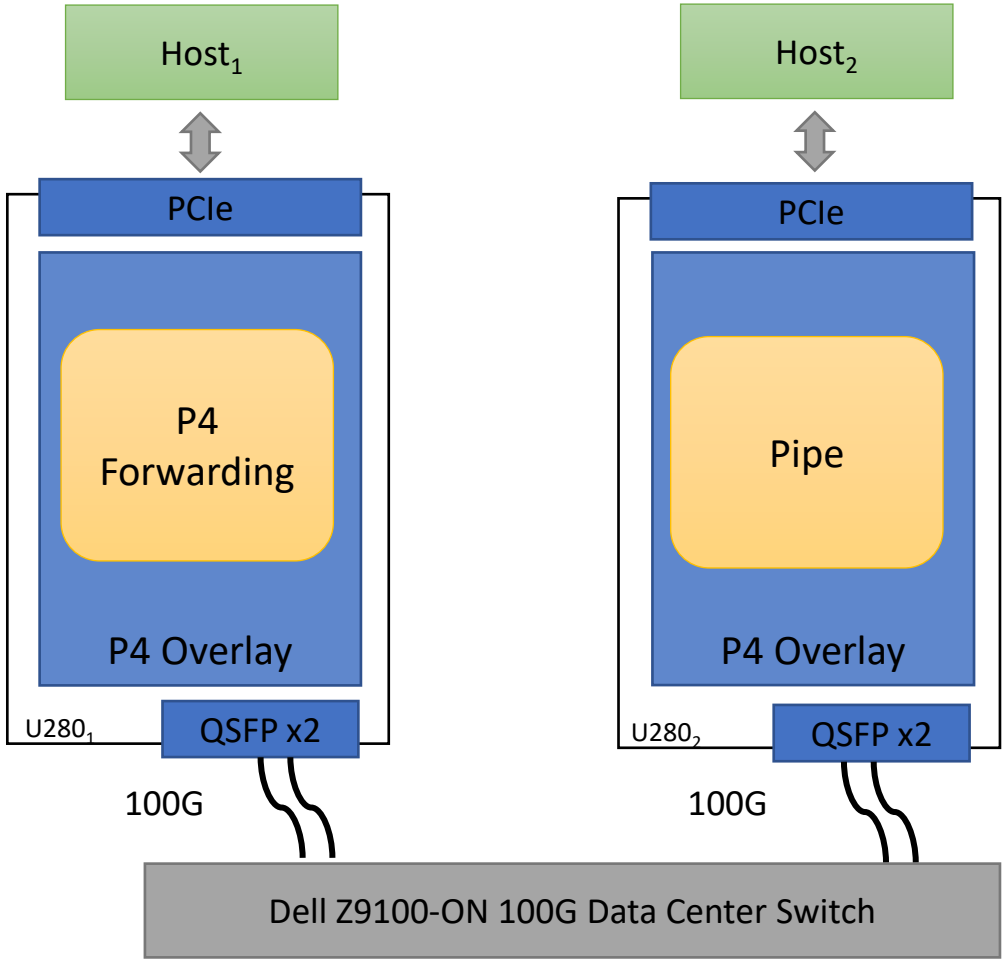
Research Directions: P4 on OCT

- Custom workflow
  - Use Vivado
  - Conventional approach: JTAG, OCT approach: PCIe
- FPGAs as SmartNICs
  - Offload network functions
  - Customize network stack
  - Support network research
- Runtime programmability
  - Live reconfiguration
  - Use of FPGA hardware features
  - Deployment network functions on the fly
  - Reduction of network downtime

Case study: L2 Forwarding

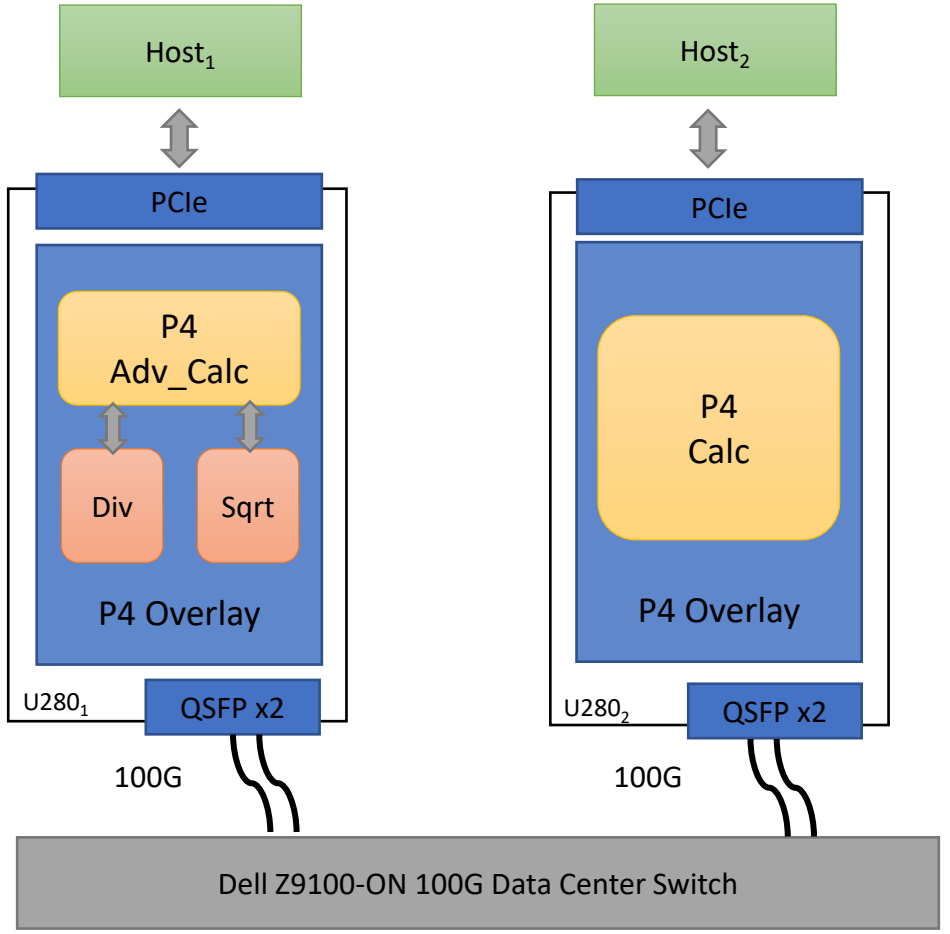
- L2 forwarding between two FPGA nodes
- Runs at 100Gbps





Implementation and Case Study

- Enabling P4 externs
- Utilizing FPGAs hardware libraries
- Providing additional computation abilities



(c)

More Tutorials and documentation

- Tutorials
- Getting started with NERC and CloudLab workflow

https://github.com/OCT-FPGA/OCT-Tutorials

- Vector addition (hello world) – https://github.com/Xilinx/Vitis_Accel_Examples
- Advanced applications (network-attached FPGAs)

Thank You

Suranga Handagala
suranga@northeastern.edu
Miriam Leeser
mel@coe.neu.edu
https://northeastern.edu/rcl

Northeastern University

BOSTON UNIVERSITY

University of Massachusetts Amherst

intel

Open Cloud Testbed
Exploring Next-Generation Cloud Platforms

NSF

MOHPCC
MASSACHUSETTS GREEN HIGH PERFORMANCE COMPUTING CENTER

NERC

CloudLab

AMD XILINX

Grant Nos:
CNS 1925658
CNS 2130891
CNS-1925464
CNS-1925504