XACC – State of the ETH Cluster

Systems@ETH zürich

Zhenhao He

Systems Group

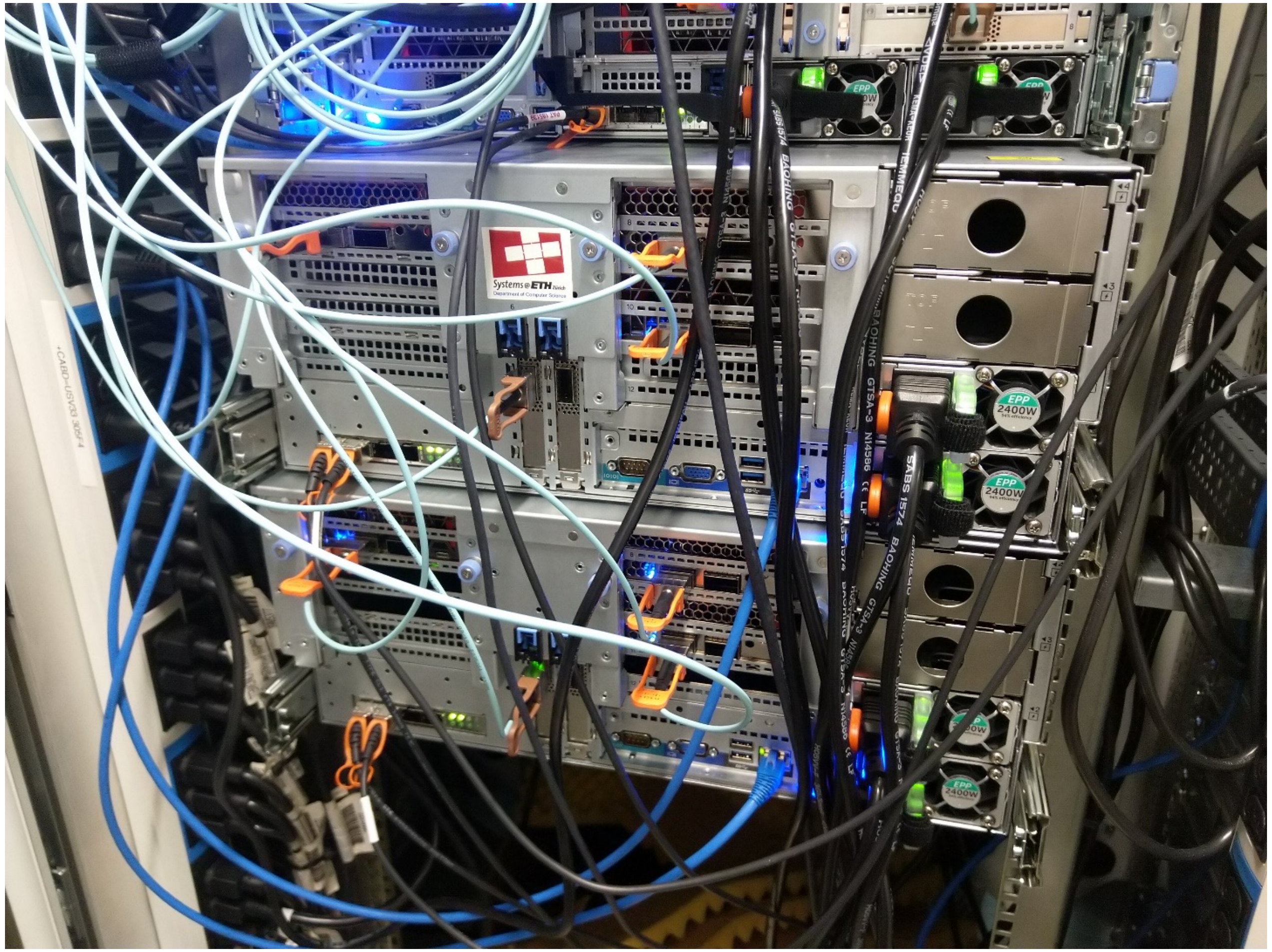
Department of Computer Science

ETH Zurich, Switzerland

Outline

- The cluster – hardware
- The cluster – software
- The cluster - set-up and use
- Issues
- Our own research
- Ideas for the future

Hardware



FPGAs deployed

- 6 Alveo 250
- 4 Alveo 280

Passive Option

Cluster (on a rack)

- 5 nodes
  - 3 x 2U
  - 2 x 4U
- 1 build node (no FPGA)
- 4 nodes with FPGAs
  - 2 x (2 x U250)
  - 2 x (2 x 280 + 1 x 250)
- 100 Gbs switch
  - For servers
  - For FPGAs (1 port)

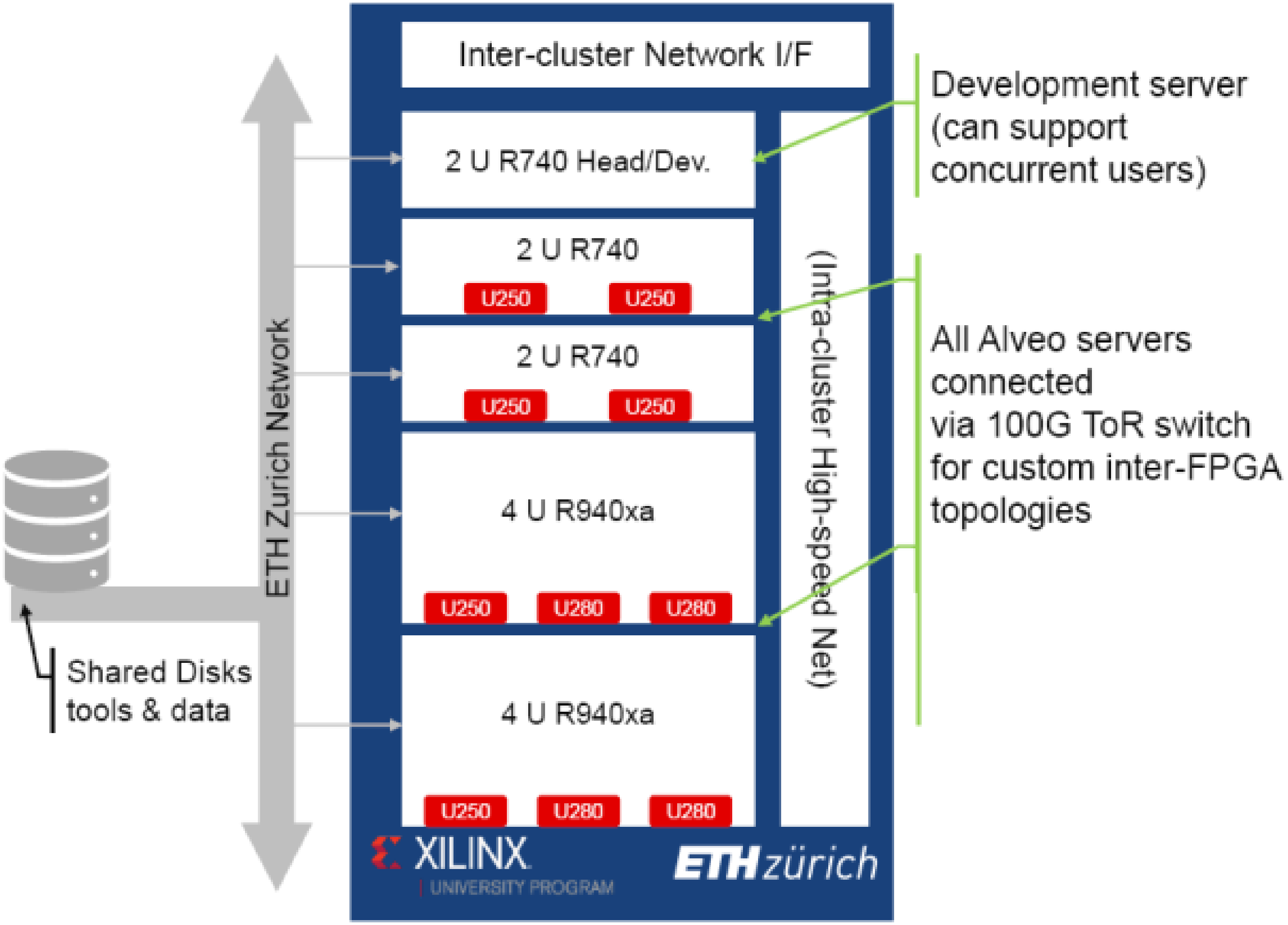


Image courtesy of Mario Ruiz

Build server (no FPGA)

- Dell Power Edge R740 (2U)
  - 2 x Intel Xeon Gold 6248 2,5 GHz, 20C/40T
  - 12 x 32 GB DDR4
  - 6 x 960 GB SSD
  - Mellanox Connect X-5, single port (100Gb)
  - Intel 10 Gbs card
- Large server for compilation, project development, and support of cluster activities
- Large enough to support many concurrent users

Nodes with 2 FPGAs

- Dell Power Edge R740 (2U)
  - 2 x Alveo U250
  - 2 x Intel Xeon Gold 6234 3,3 GHz, 8C/16T
  - 12 x 32 GB DDR4
  - 2 x 96GB SSD
  - 2 x Mellanox Connect X-5, single port (100Gb)
  - Intel 10 Gbs card

Nodes with 3 FPGAs

- Dell Power Edge R940 (4U)
  - 2 x 2 x 2 x Intel Xeon Gold 6234 3,3 GHz, 8C/16T
  - 24 x 16 GB DDR4
  - 2 x 96GB SSD
  - 2 x Mellanox Connect X-5, single port (100Gb)
  - Intel 10 Gbs card

Network configuration

- Each FPGA has two network ports
  - One connected to ToR switch
  - One connected to another FPGA

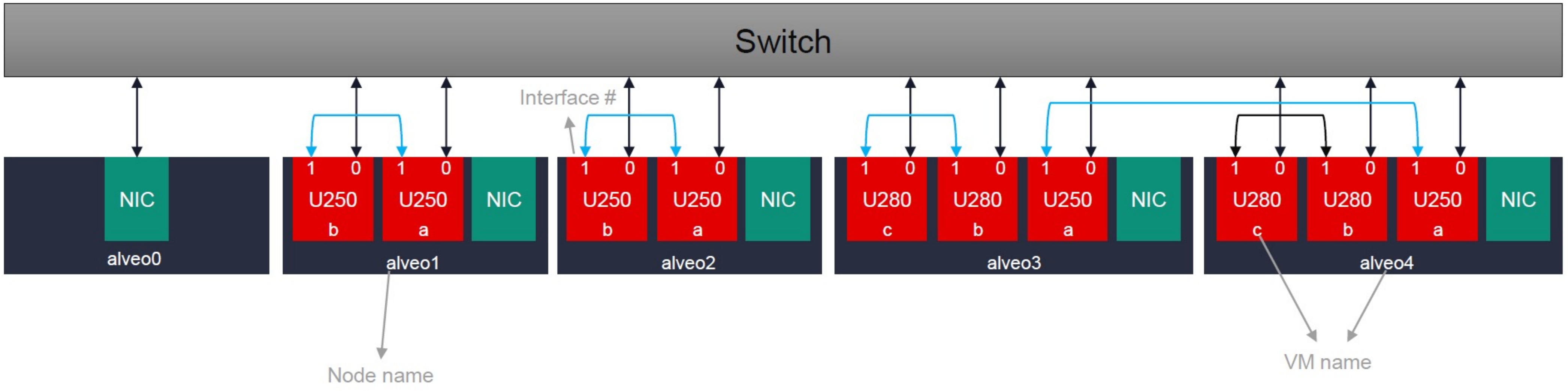


Image courtesy of Mario Ruiz

Software

Software in general

- Build server intended for general use, compilation, builds, etc.
- Servers with FPGAs only intended to run applications and for testing/debugging purposes (XILINX tools Chipscope, Vitis profiler). Not intended for building projects

- EXTRA SYSTEMS
  - PYNQ
  - InAccel's Coral (FPGA Cluster tool)

Configuration on nodes with FPGAs

- Hypervisors:
  - OS: Debian 10
  - virtualization technology: KVM/QEMU
  - PCI passthrough: Alveo FPGA, Mellanox ConnectX-5
- 10x single XRT FPGA Virtual Machines (one per FPGA):
  - OS: Ubuntu 18.04
  - shared network (iSCSI) disk mounted, with the Xilinx tools:
    - Vitis (2019.2, 2020.1)
    - Vivado (2019.2, 2020.1)
  - Home directory mounter
  - Shared disk
- 4x multi-FPGA bare-metal Virtual Machines
  - “Raw” mode, no Xilinx Shell loaded
  - ETH internal use & maintenance

Configuration machines and VMs

- Not all VMs has the PCIe passthrough of 100 Gbps NIC

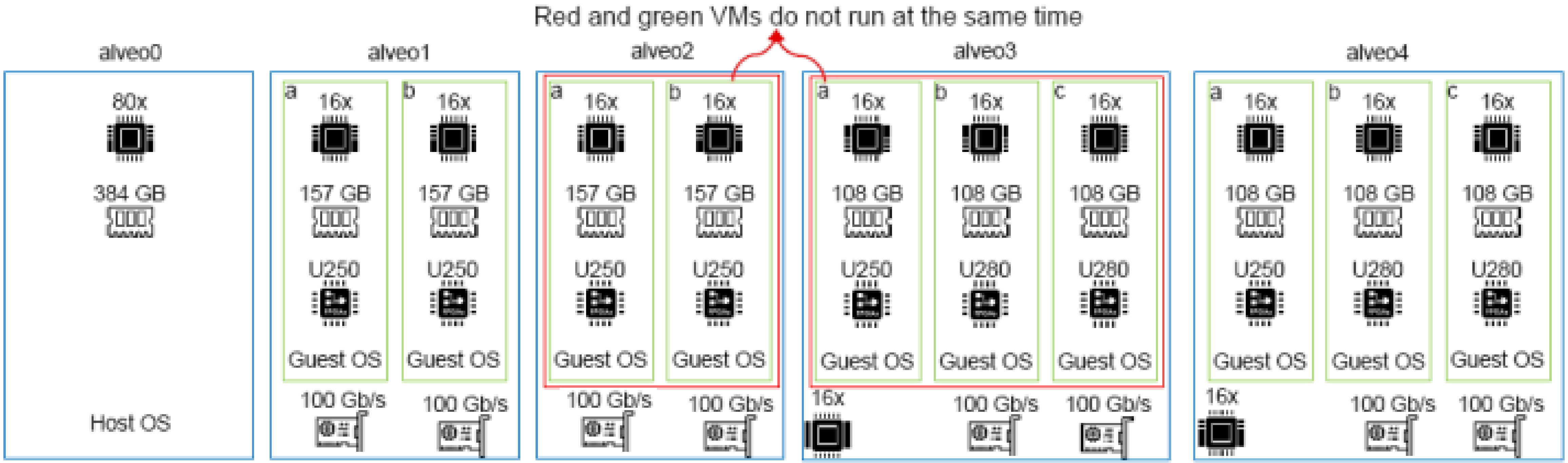


Image courtesy of Mario Ruiz

Shell & XRT

- XRT 2.8.743

- On Alveo 0 (build server):
  - Vitis 2019.2, 2020.1, 2020.2
  - Vivado 2019.2, 2020.1, 2020.2

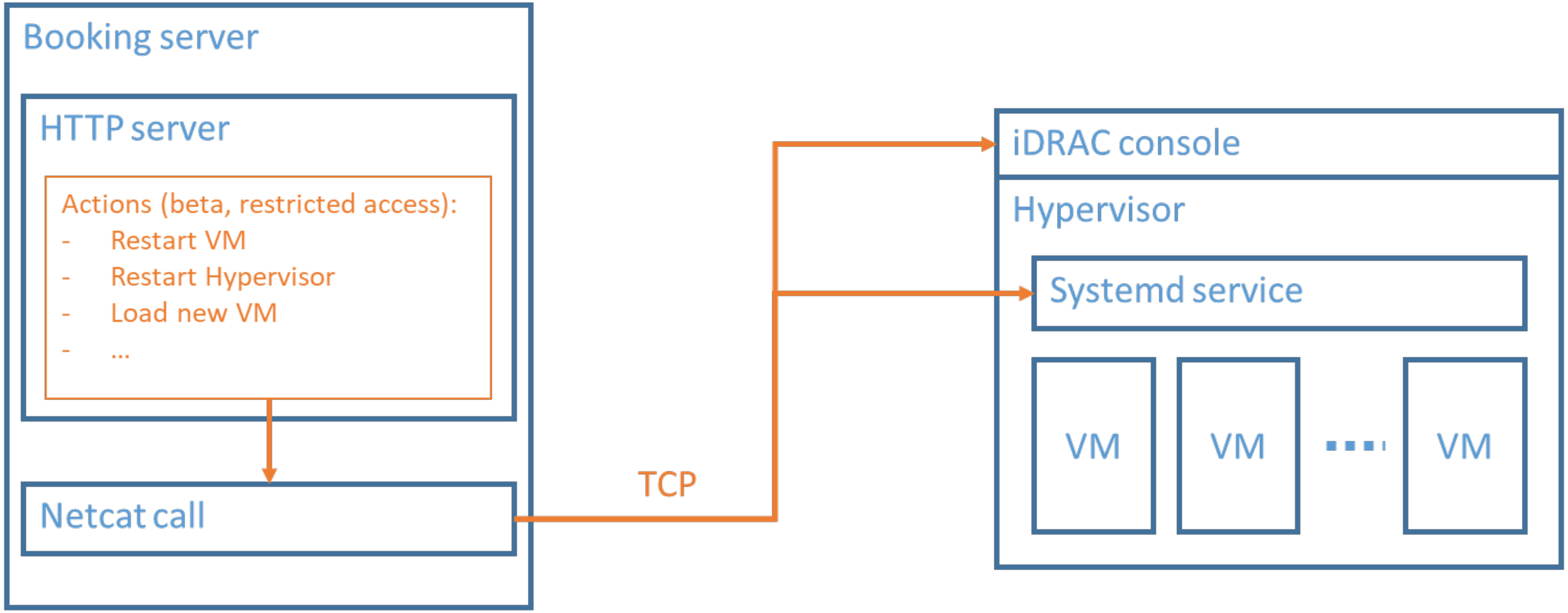
- On servers with FPGAs:
  - Provide shell with network support
  - U250: xilinx_u250_gen3x16_xdma_shell_3_1
  - U280: xilinx_u280_xdma_201920_3

Set-up and use

Users

- Access requires registration
  - ETH users contact Gustavo Alonso
  - All others through Xilinx (XACC program)
  - Users get guest account at ETH (renewable)
- Currently operating on trust and good faith
  - Slowly setting up some basic rules of operation
- As of December 2020
  - 29 institutions, 15 countries
  - 75 registered users

Booking system



Integrated Dell Remote Access Controller (iDRAC): Allows performing shutdown, cold reboot of the HV

Open-source: We can share the code of all the infrastructure. Contributions are welcomed

Booking system

- Features that work well:
  - Reserving a specific VM for a specific period is the main goal, and it works well
  - During a reservation, only the selected user is able to connect to the VM
  - Some users (beta, restricted access) can load different VMs via the dashboard.
- Features that do not work well:
  - Users must manually switch back to the standard VM when they are done (if using a different VM).
  - In some cases (when VM is not responding), the system will wait forever to perform the actions (need some better detection mechanism -- e.g., timeout)
  - Lacking statistics, monitoring, etc.
  - Users need to book VMs even if no FPGA experiment is needed, e.g., network experiment with 100 Gbps commodity NIC

Issues

Issues

- Minor (hardware)
  - U250 shifted from chassis, needed elongated USB cable
  - Some boards run into problems due to misconfigurations
  - JTAG cable for U280 is internal (requires to open the server)
- More involved:
  - Difficult to support different shells
    - Manual process and cold reboots needed
    - FPGA not configurable through PCI (we use JTAG, going back to Vitis requires reboot)
  - FPGAs disappear from PCIe bus
    - Must be physically unplugged form server and plugged back
  - Heterogeneity of shells and capabilities

Use cases

- The use case patterns are not clear, which limits the effectiveness of sharing and the booking system
  - Single FPGA usage vs multi-FPGA usage
  - Access to the entire cluster
  - Research vs computation
- Some use cases require access to the “raw” FPGA
  - Important for a lot of research on systems
  - Currently very cumbersome to do

Our own research

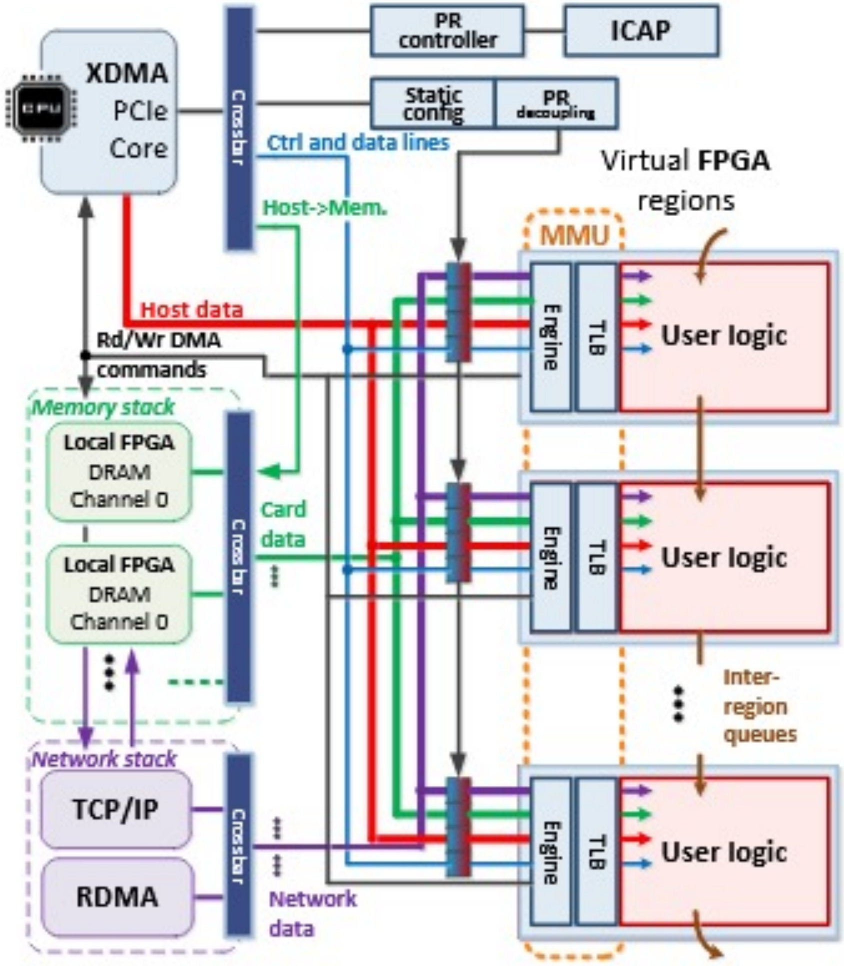
Research on the cluster

- Systems:
  - Coyote: an OS for the FPGA
- Infrastructure:
  - EasyNet: 100 Gbs TCP/IP stack for Vitis
- Applications:
  - Distributed Recommendation Inference

Coyote

- Multiple user regions (6 to 10)
- RDMA/TCP network stack
- Unified memory space host-FPGA
- Virtual memory
- Multi-user memory management on FPGA

Do OS abstractions make sense on FPGAs?, Dario Korolija, Timothy Roscoe, and Gustavo Alonso, OSDI 2020

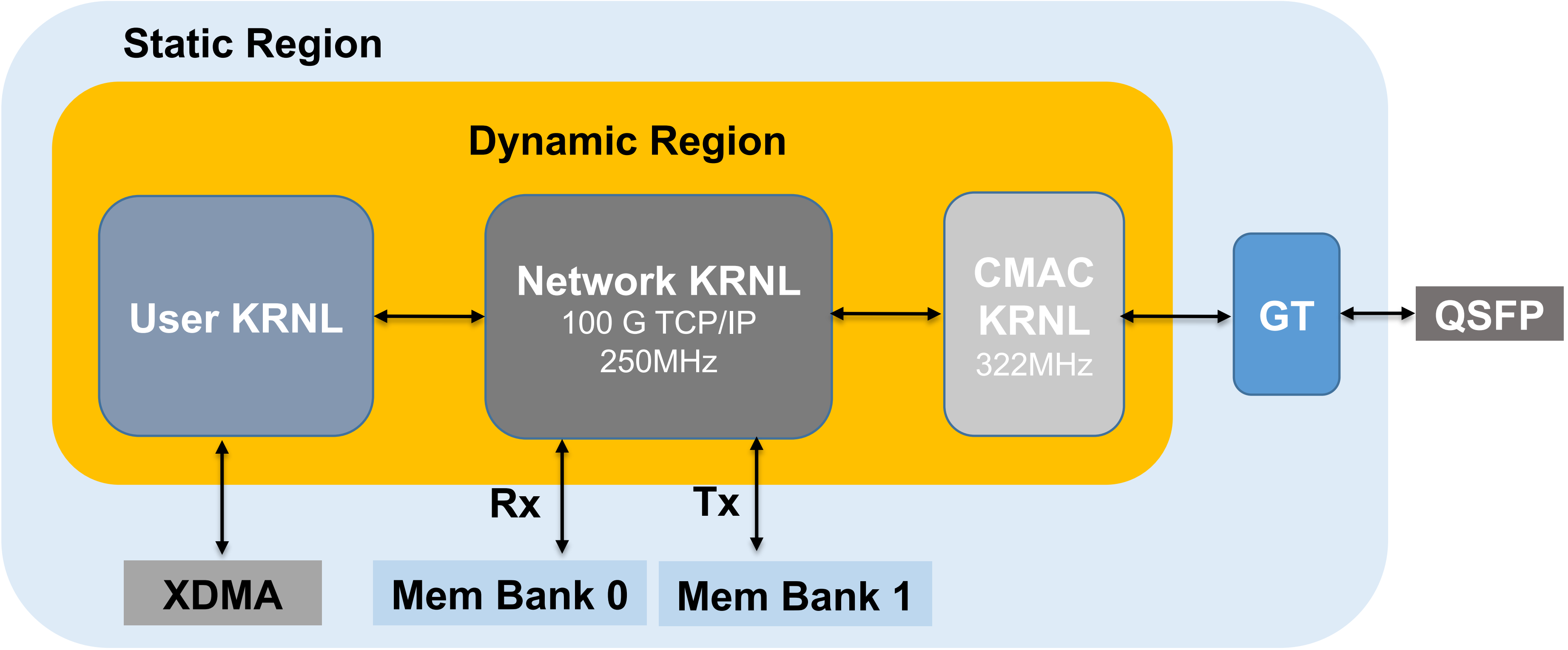


EasyNet’s Goal

- **Integrate a 100 Gbps TCP/IP stack into Vitis platform**
  - Take advantage of HLS
  - Abstract network data movement
- **Provide higher level API for communication**
  - Point-to-point communication
  - Collective communication
  - Easy to instantiate

Overall Architecture

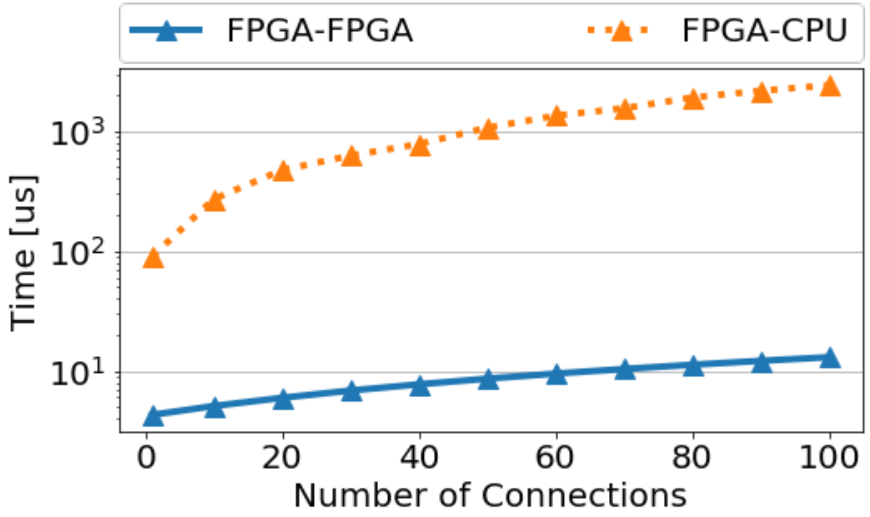
- CMAC: Ethernet subsystem, board specific
- Network: TCP/IP stack with streaming control and data interfaces
- User: Customized unit for application



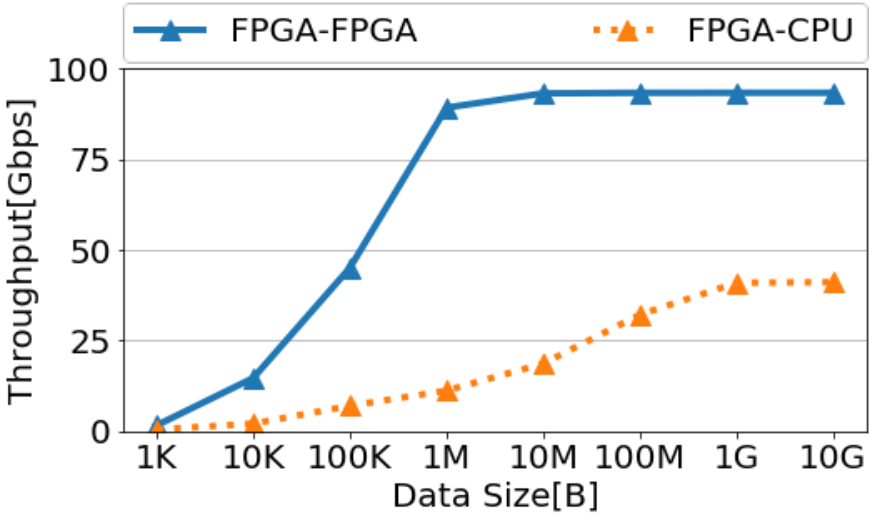
Performance – Latency and Throughput

- Latency
  - RTT : FPGA-FPGA 5 us VS FPGA-CPU 90 us

- Throughput
  - FPGA send & FPGA receive saturates 100 Gbps with 1 MB data
  - FPGA send & CPU receive on a single core achieves 46 Gbps



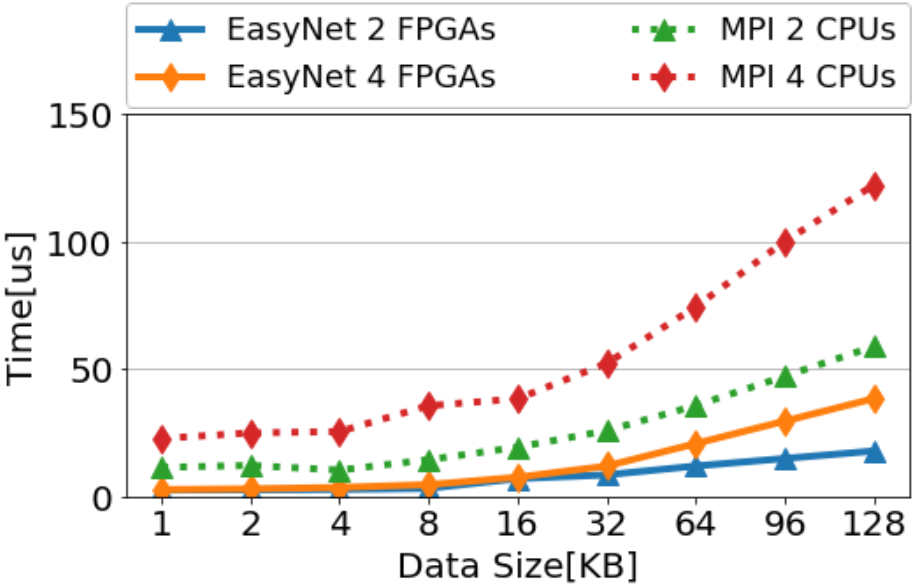
Open Connection Time



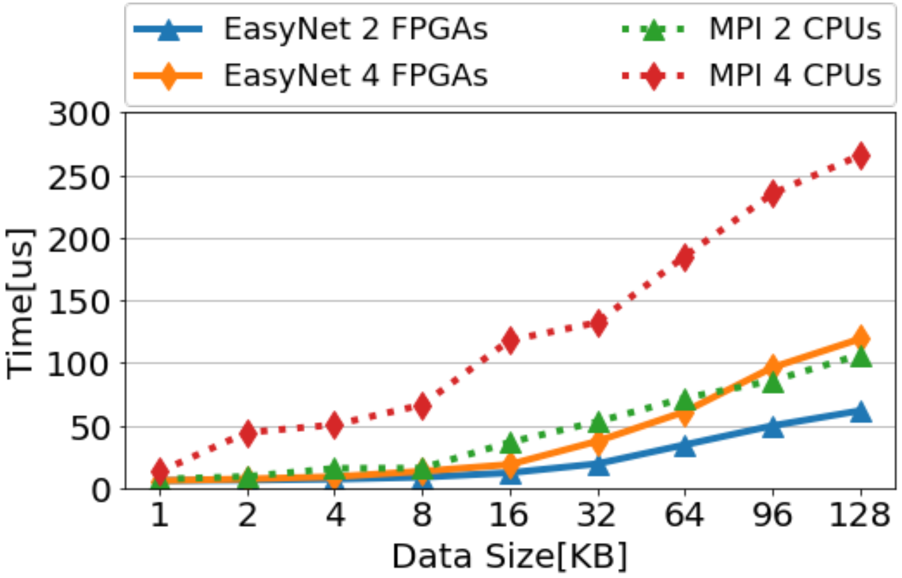
Send & Recv Primitive Throughput

Performance – Collective Primitives

- Broadcast and all-reduce on 4 FPGAs
- Compared with OpenMPI on 4 CPUs
- FPGA implementation achieves lower latency



Broadcast



All - Reduce

Status

- EasyNet available on XACC ETHZ cluster and as open source
  - Vitis 2019.2 (stable release) and 2020.1 (initial tests)
  - Running on U280s (soon U250s once all shells are updated)
- EasyNet presentation at FPL'21: Thursday, Session 3B: Memory, Network & Streams
- Currently developing collective primitives
  - Interaction with Xilinx on the topic

Distributed Recommendation Inference on FPGA Clusters

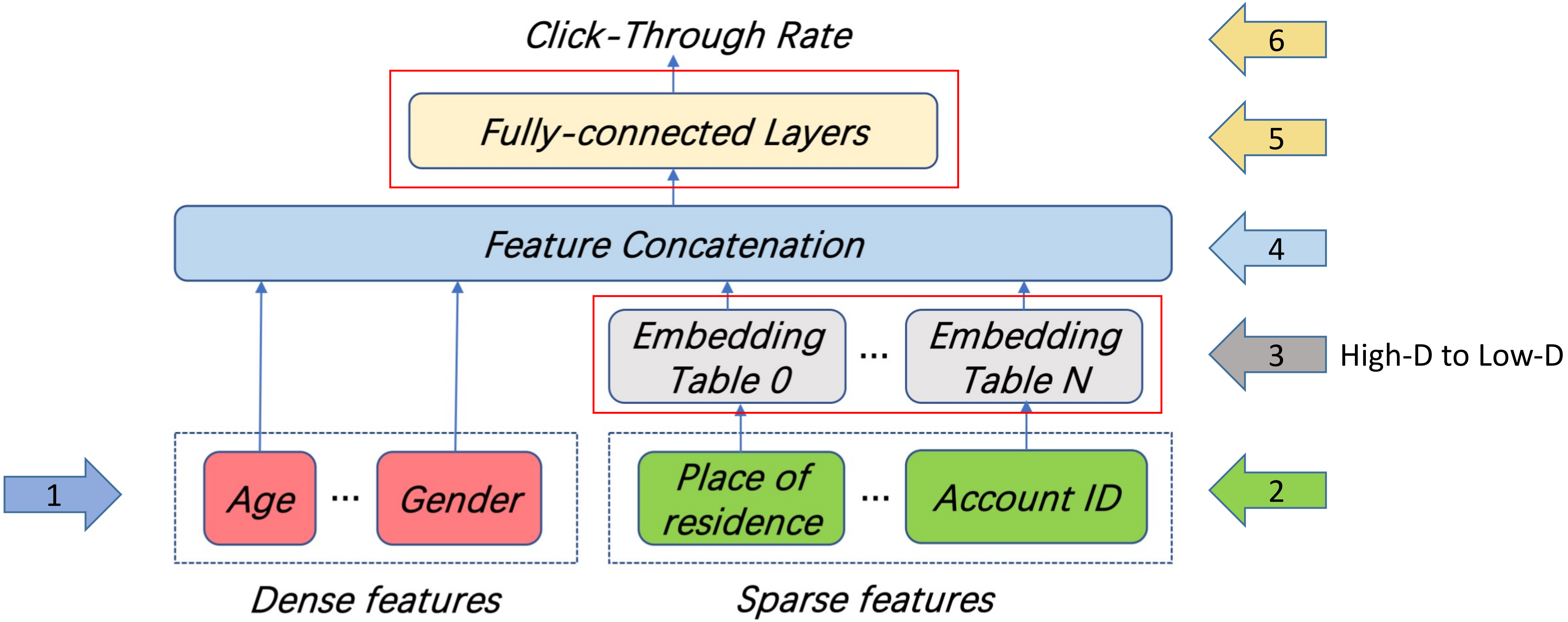


Fig. 1. High-level architecture of a typical deep recommendation model.

Target: design an FPGA-based recommendation inference system that tackle both the memory and computation bottlenecks

- Main idea
  - Accelerate both embedding lookup and computation
  - Match the speed in different stages

Method: take advantage of the strengths of FPGA cluster

I. EasyNet: TCP/IP Network stack for cluster[1]

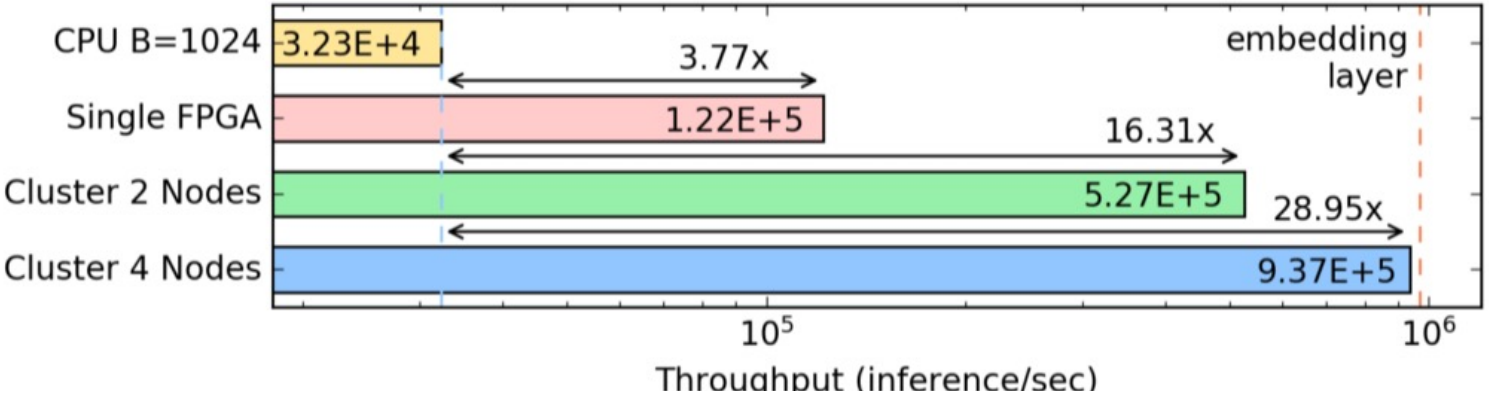
II.Use HBM for embedding lookup on single node

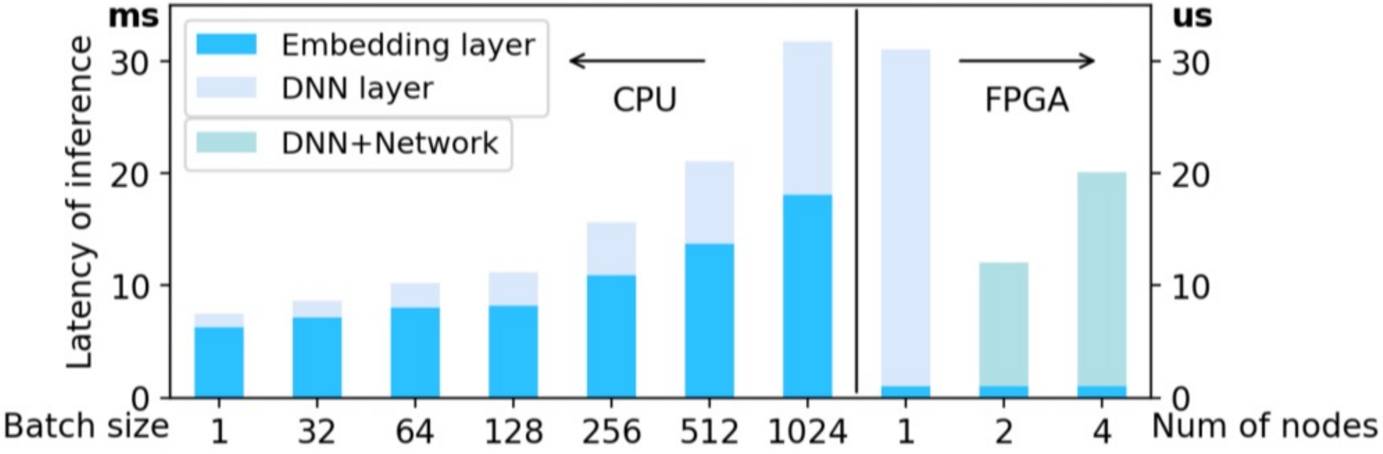
III.Partition computation among nodes

[1] He, Z., et al. (2021). EasyNet: 100 Gbps Network for HLS. 31th International Conference on Field Programmable Logic and Applications(FPL), IEEE. https://github.com/fpgasystems/Vitis_with_100Gbps_TCP-IP.git

Evaluation

- Presentation at FPL'21: Friday, Session 5A: Accelerated Machine Learning (2)





Ideas for the future

Working on a complete platform

- Instead of making FPGAs available ...
- ... make an ecosystem available:
    - Infrastructure from the FPGA (networking, storage, management)
    - Tools on FPGA clusters (MPI library, distributed coordination)
    - Basic applications (Key value store, ML libraries, ...)

- In the cloud, not even the large companies build everything from scratch: deep stack of open source applications and infrastructure
- Potential goal: foster the creation of such an open source stack for clusters of FPGAs