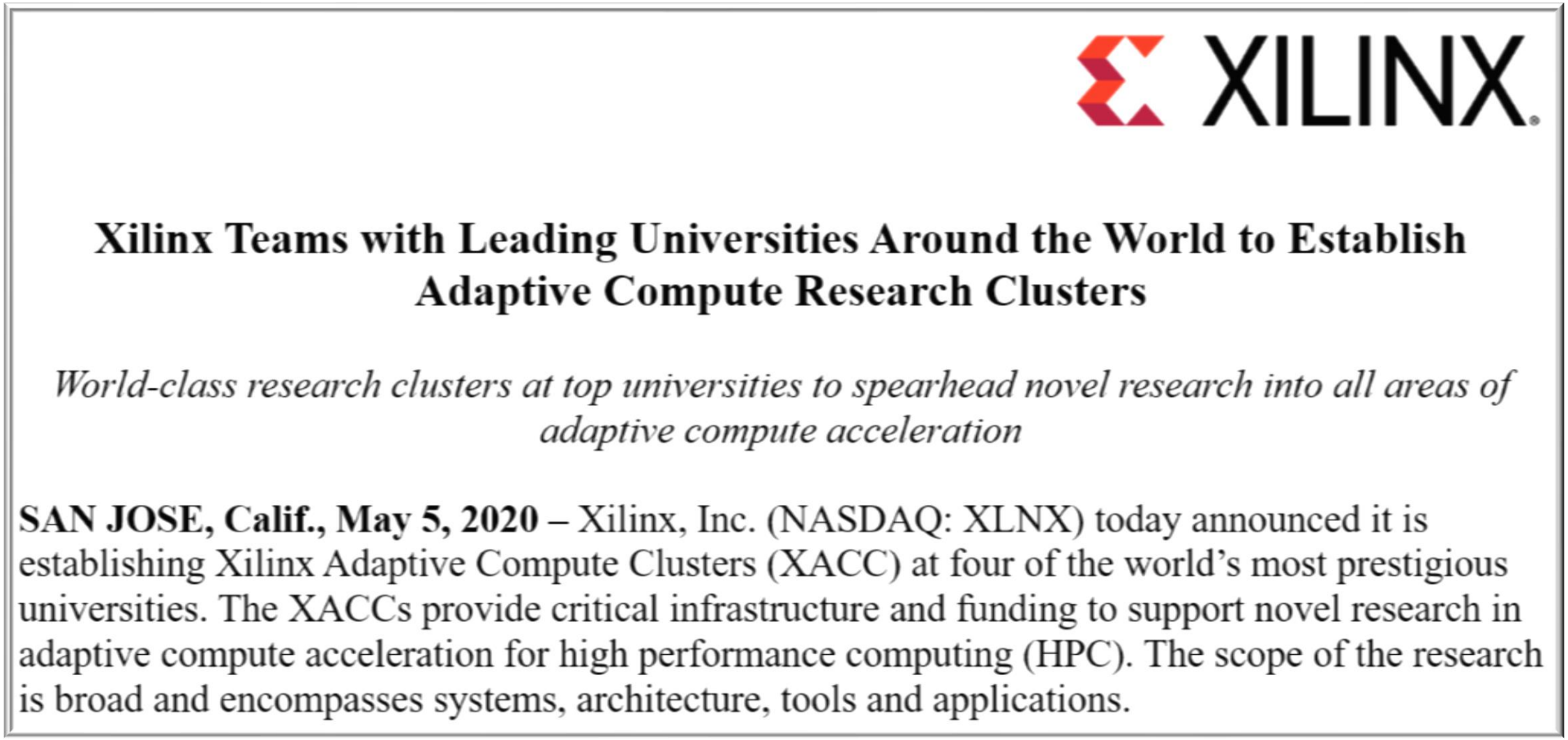
XILINX

Xilinx Adaptive Compute Clusters

Xilinx University Program

Xilinx Adaptive Compute Clusters



XILINX

Xilinx Teams with Leading Universities Around the World to Establish Adaptive Compute Research Clusters

World-class research clusters at top universities to spearhead novel research into all areas of adaptive compute acceleration

SAN JOSE, Calif., May 5, 2020 – Xilinx, Inc. (NASDAQ: XLNX) today announced it is establishing Xilinx Adaptive Compute Clusters (XACC) at four of the world's most prestigious universities. The XACCs provide critical infrastructure and funding to support novel research in adaptive compute acceleration for high performance computing (HPC). The scope of the research is broad and encompasses systems, architecture, tools and applications.

Xilinx Adaptive Compute Clusters (XACC)

XILINX
UNIVERSITY PROGRAM

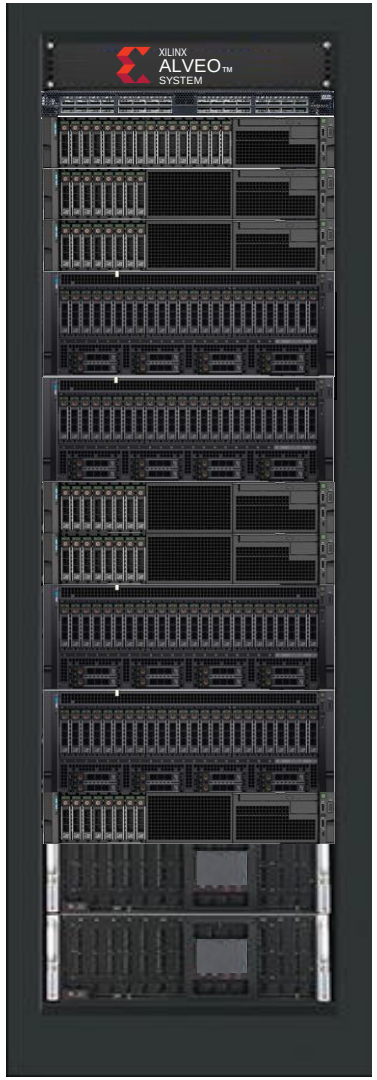
- CTO initiative
- XUP and Xilinx Research Labs are leading it

XILINX

Research Areas

- Adaptive Compute Acceleration
- High Performance Computing (HPC)
- Machine Learning
- Database Acceleration
- Energy Efficiency
- Compilers
- IoT
- Computer architecture

XACC Hardware



Alveo U250

Alveo U280

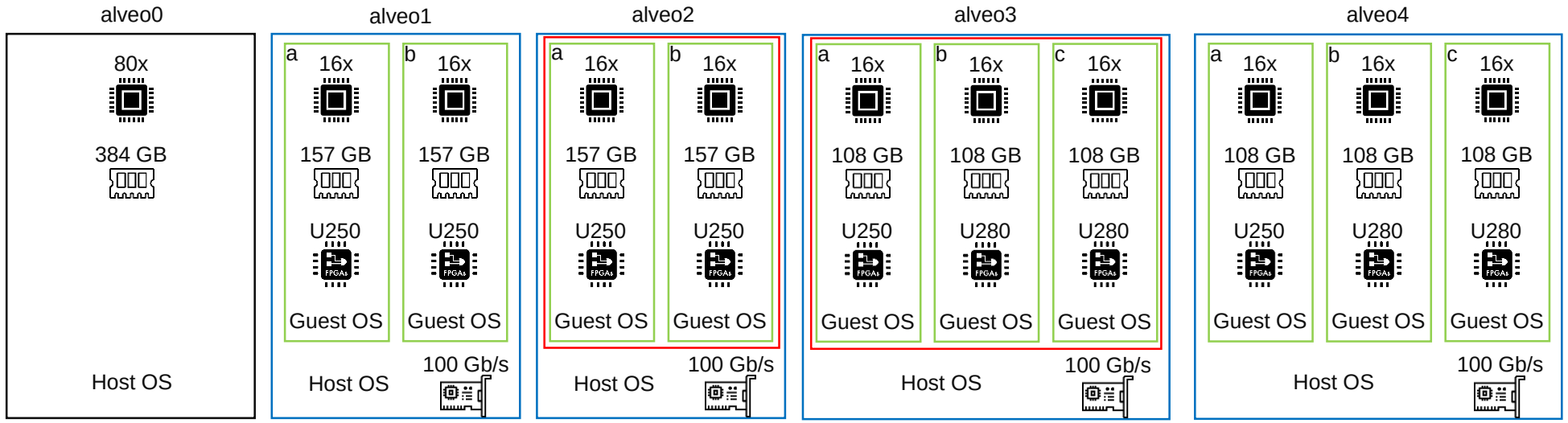
- XACC hardware consists of:
  - Compute and Alveo nodes
  - 100G network
- Initially Alveo U250 and U280 (HBM)
  - Versal will be added in later deployment
- Community hub for researchers
  - Support from in-house Xilinx research groups
  - Reproducible results & experiments

ETH Zurich XACC

XILINX.

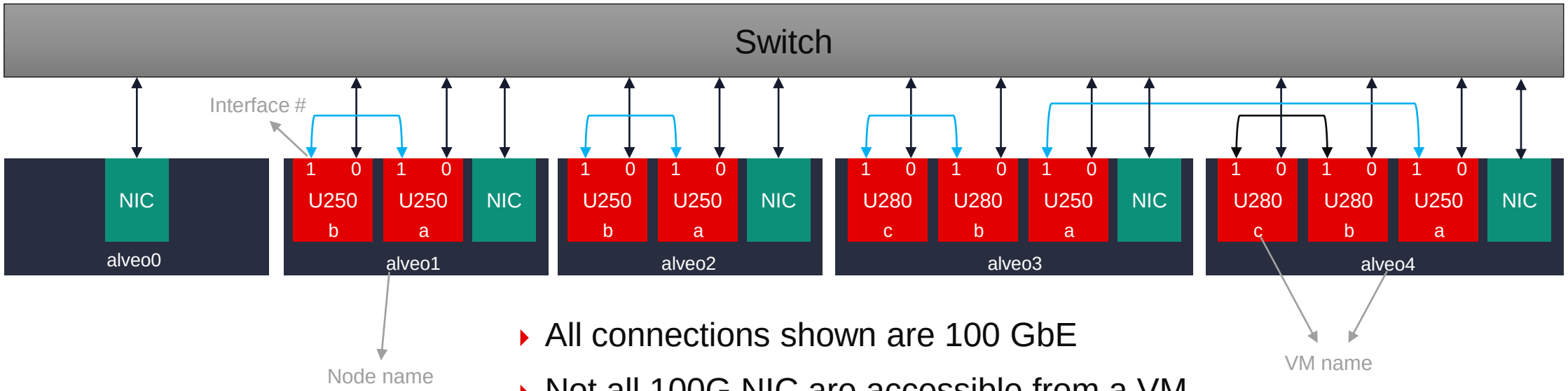
Cluster Organization

- Development server
  - Any user can access it
  - Use your own if possible
- KVM to virtualise Alveo nodes
  - Static VMs
  - One Alveo card per VM
  - Special VM for multiple Alveo cards
  - Users reserve VM(s)
    - Exclusive usage



XILINX.

Network Organisation



- All connections shown are 100 GbE
- Not all 100G NIC are accessible from a VM
- IPs available 10.1.212.160 to 10.1.212.200
- VMs can access this infrastructure using the 10GbE NIC

XILINX.

System Overview - Software

- Ubuntu 18.04 in host and guest OS
- Vitis/Vivado
  - 2020.1; 2020.2; 2021.1
  - Installed natively on development server
  - VM can access them from shared network disk from Alveo nodes
    - Tools expected to be used to debug and profiling only (no Alveo builds)
- XRT (Xilinx Run Time)
  - Multiple versions on development server
  - 2.9.317 in deployment servers
- User home directory mounted in all VMs and dev server (NFS)
- Shared disk for user data

Hardware – intended use

- Development server
  - Intended for Alveo builds
  - Expect (most) users will run builds on their own local servers, but can access and use this server
- Alveo nodes
  - Xilinx tools installed, but intended for test/debug only (e.g. Chipscope, Vitis profiler)
  - Not intended for Alveo builds, maximize utilization

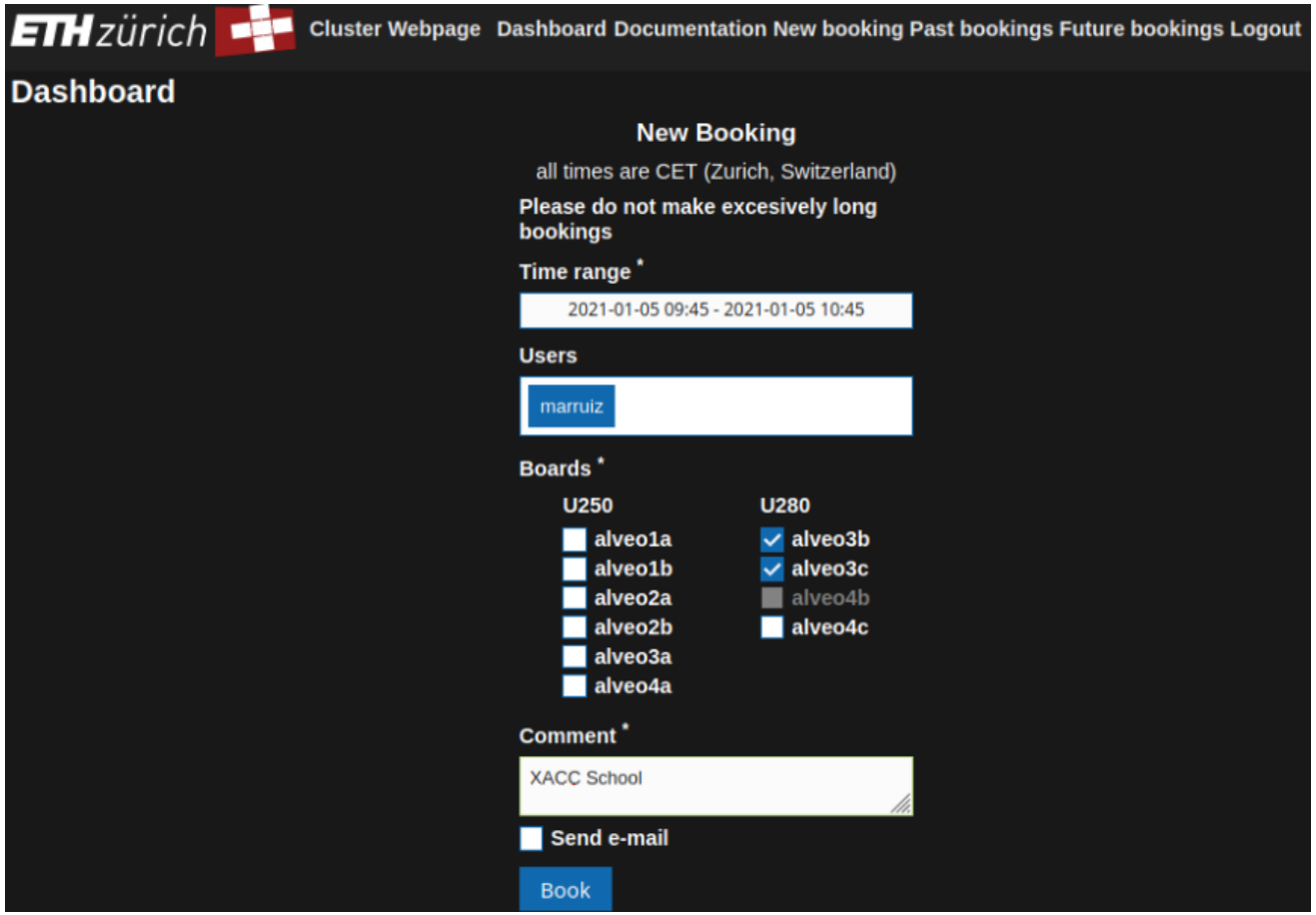
Testing

- XRT
  - Check XRT tools installed
- xbutil can be used to do test basic functionality of Alveo hardware
  - Readback board status
  - DMA test available for all boards which loads simple design to PL to test bandwidth

```
xbutil scan
xbutil query -d 0
xbutil validate
xbutil dmatest -d 0 -b 0x1000
```

Booking System

- User gets guest account
  - User can book time on a VM
- Multiple VM can be reserved
- Booking will reserve machines
  - Exclusive VM access
  - Single or multiple users

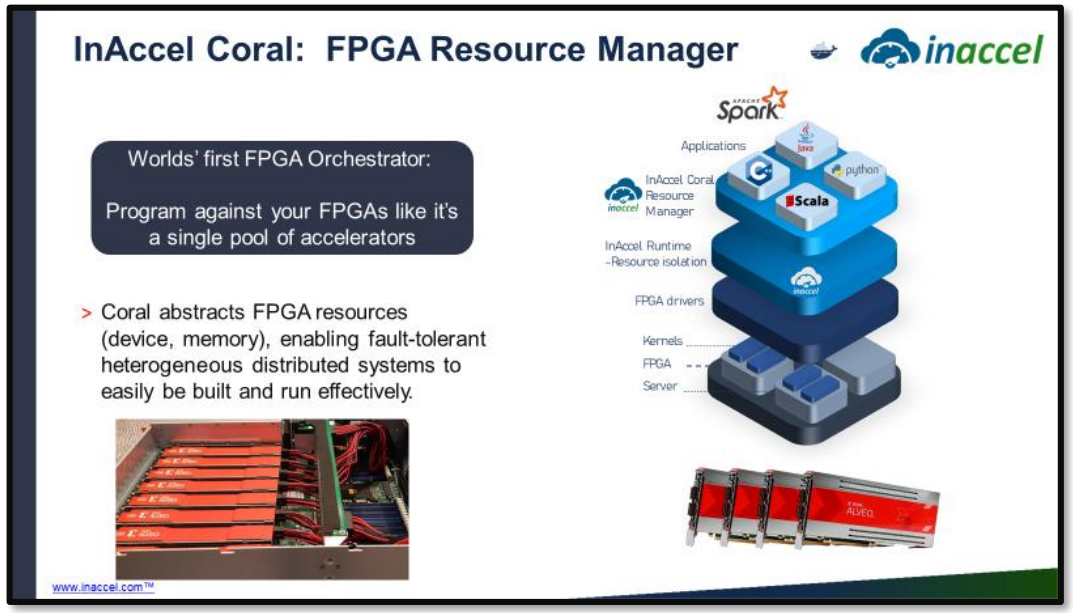


XILINX.

ETH Zurich Extra systems

XILINX.

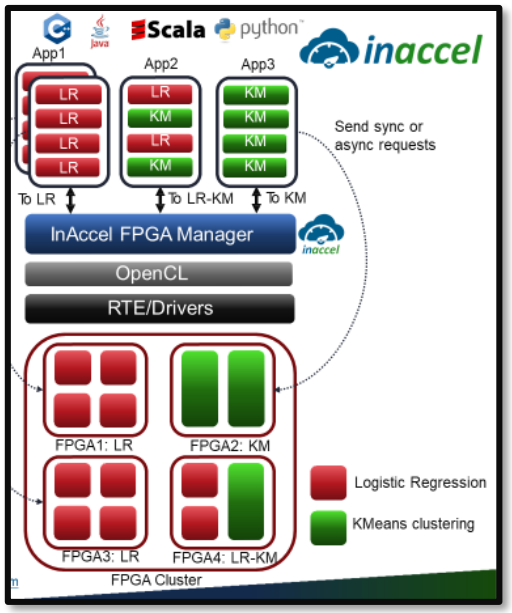
InAccel's Coral: *Heterogeneous Accelerator Orchestration*



Now available free at ETH Zurich XACC

Works on top of XRT

InAccel is keen to support Universities
https://inaccel.com/university/



Makes accelerators accessible to all classes of users

XILINX.

PYNQ

- PYNQ via Conda in a shared disk
  - User can install extra packages
  - Dask for distributed computation

PYNQ

```python
# device initialization
import pynq
ol = pynq.Overlay("intro.xclbin")
vadd = ol.vadd_1


# buffers allocation
size = 1024*1024
in1 = pynq.allocate((1024, 1024), 'u4')
in2 = pynq.allocate((1024, 1024), 'u4')
out = pynq.allocate((1024, 1024), 'u4')



# send data from host to FPGA
in1.flush()
in2.flush()
# run kernel
vadd.call(in1, in2, out, size)



# receive data from FPGA to host
out.invalidate()
```

OpenCL

```
auto devices = xcl::get_xil_devices();
auto fileBuf = xcl::read_binary_file(binaryFile);
cl::Program::Binaries bins{{fileBuf.data(), fileBuf.size()}};
OCL_CHECK(err, context = cl::Context({device}, NULL, NULL, NULL, &err));
OCL_CHECK(err, q = cl::CommandQueue(context, {device}, CL_QUEUE_PROFILING_ENABLE, &err));
OCL_CHECK(err, cl::Program program(context, {device}, bins, NULL, &err));
OCL_CHECK(err, krnl_vector_add = cl::Kernel(program, "vadd", &err));



std::vector<int, aligned_allocator<int>> source_in1(DATA_SIZE);
std::vector<int, aligned_allocator<int>> source_in2(DATA_SIZE);
std::vector<int, aligned_allocator<int>> source_hw_results(DATA_SIZE);
OCL_CHECK(err, l::Buffer buffer_in1(context,
    CL_MEM_USE_HOST_PTR | CL_MEM_READ_ONLY, vector_size_bytes,
    source_in1.data(), &err));
OCL_CHECK(err, cl::Buffer buffer_in2(context,
    CL_MEM_USE_HOST_PTR | CL_MEM_READ_ONLY,  vector_size_bytes,
    source_in2.data(), &err));
OCL_CHECK(err, cl::Buffer buffer_output(context,
    CL_MEM_USE_HOST_PTR | CL_MEM_WRITE_ONLY, vector_size_bytes,


OCL_CHECK(err, err = q.enqueueMigrateMemObjects({buffer_in1, buffer_in2},
                                                0 /* 0 means from host*/));



int size = DATA_SIZE;
OCL_CHECK(err, err = krnl_vector_add.setArg(0, buffer_in1));
OCL_CHECK(err, err = krnl_vector_add.setArg(1, buffer_in2));
OCL_CHECK(err, err = krnl_vector_add.setArg(2, buffer_output));
OCL_CHECK(err, err = krnl_vector_add.setArg(3, size));
// send data here
OCL_CHECK(err, err = q.enqueueTask(krnl_vector_add));
// retrieve data here
q.finish();



OCL_CHECK(err, err = q.enqueueMigrateMemObjects({buffer_output},
                                                CL_MIGRATE_MEM_OBJECT_HOST));
```

PYNQ™ Alveo Edition

XILINX

Inter FPGA

- EasyNet
- VNx
- ACCL
- PCIe Peer-to-Peer (upon request)

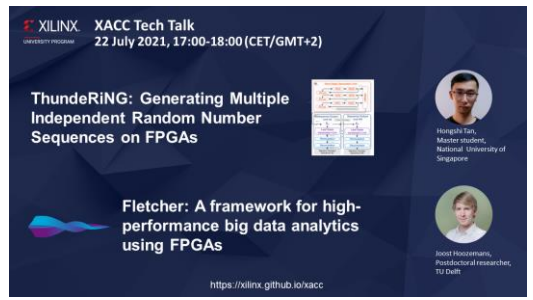
Join XACC

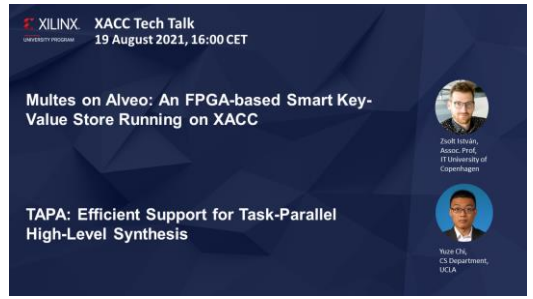
XILINX.

XACC Tech Talks

- Virtual Talks
  - 2x 30minute per session
- Any topics related to XACC research
  - Systems, architectures, tools, applications
- Summer series available on-demand
  - 7x sessions, 14x talks
- Contact us to give a talk

Available on-demand:
https://xilinx.github.io/xacc

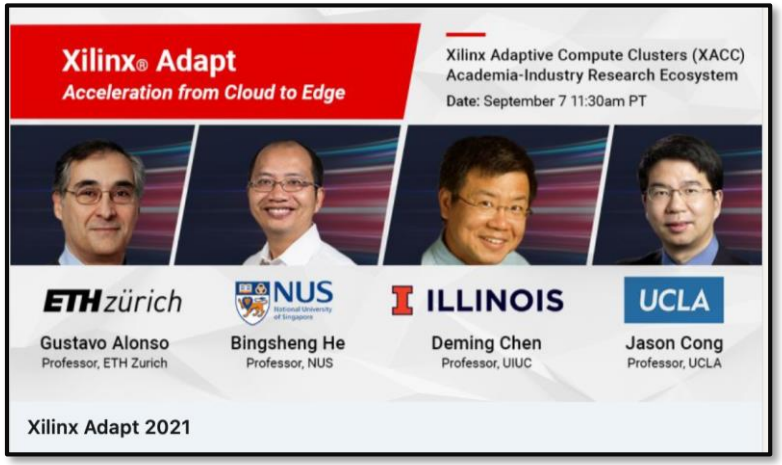




XACC Presentation at Xilinx Adapt 2021

Part of the XUP session

Session
Design Rational of Two Generations of AI Engines
Software-Defined Radio with the Open-Source RFSoC-PYNQ Framework
Xilinx Adaptive Compute Clusters (XACC) Academia-Industry Research Ecosystem
Python Productivity for the Kria SOM: PYNQ, Ubuntu, and the New Composable Overlay
Announcement of the 2021 Awards for the Xilinx Women in Technology University Grants



Excellent attendance

Available on YouTube https://youtu.be/ousTRtD4m2Q

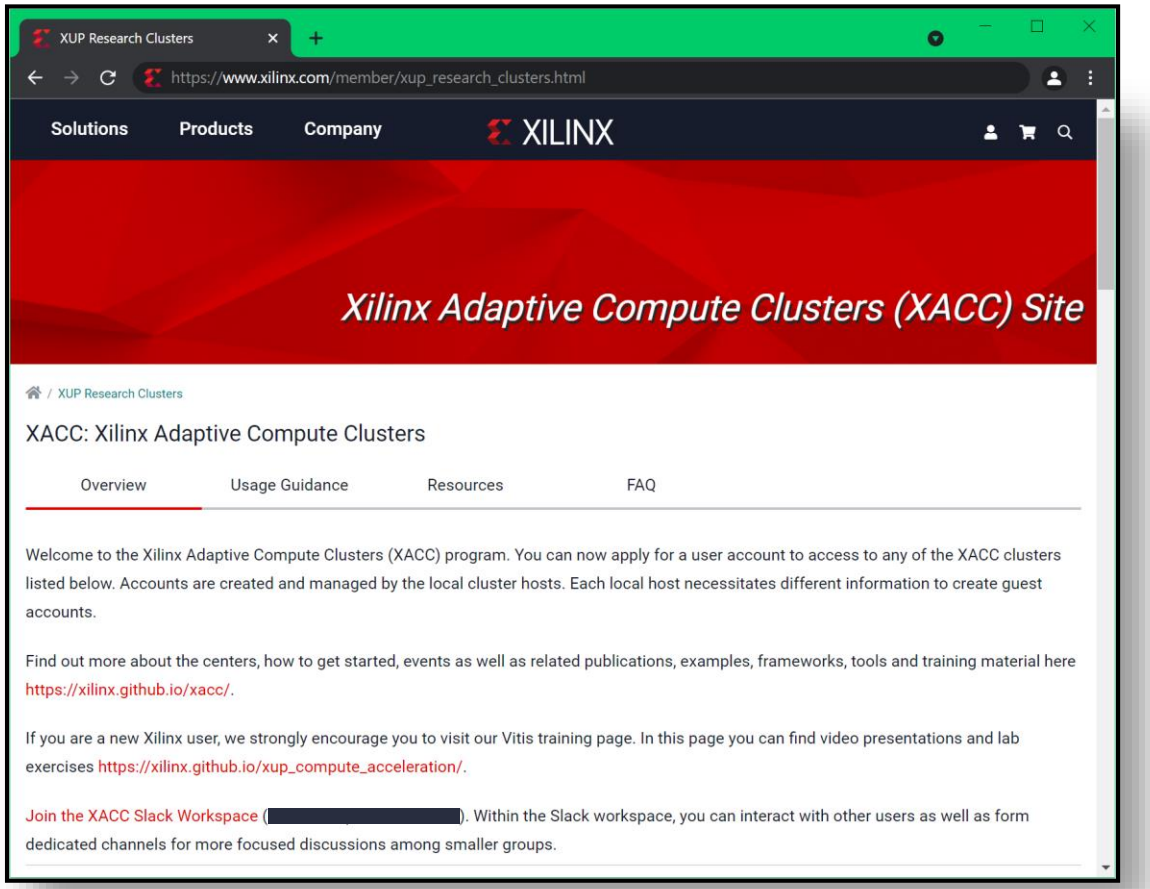
XILINX

Excellence in Adaptive Computing Awards

XILINX

XACC lounge site

- Documentation
  - Getting Started Guide(s)
  - Acknowledgements section
  - Link to useful resources
  - User Guidance
- Access to exclusive content
  - Contact details
  - User meeting recordings
  - Slack Workspace link
- FAQ



XILINX

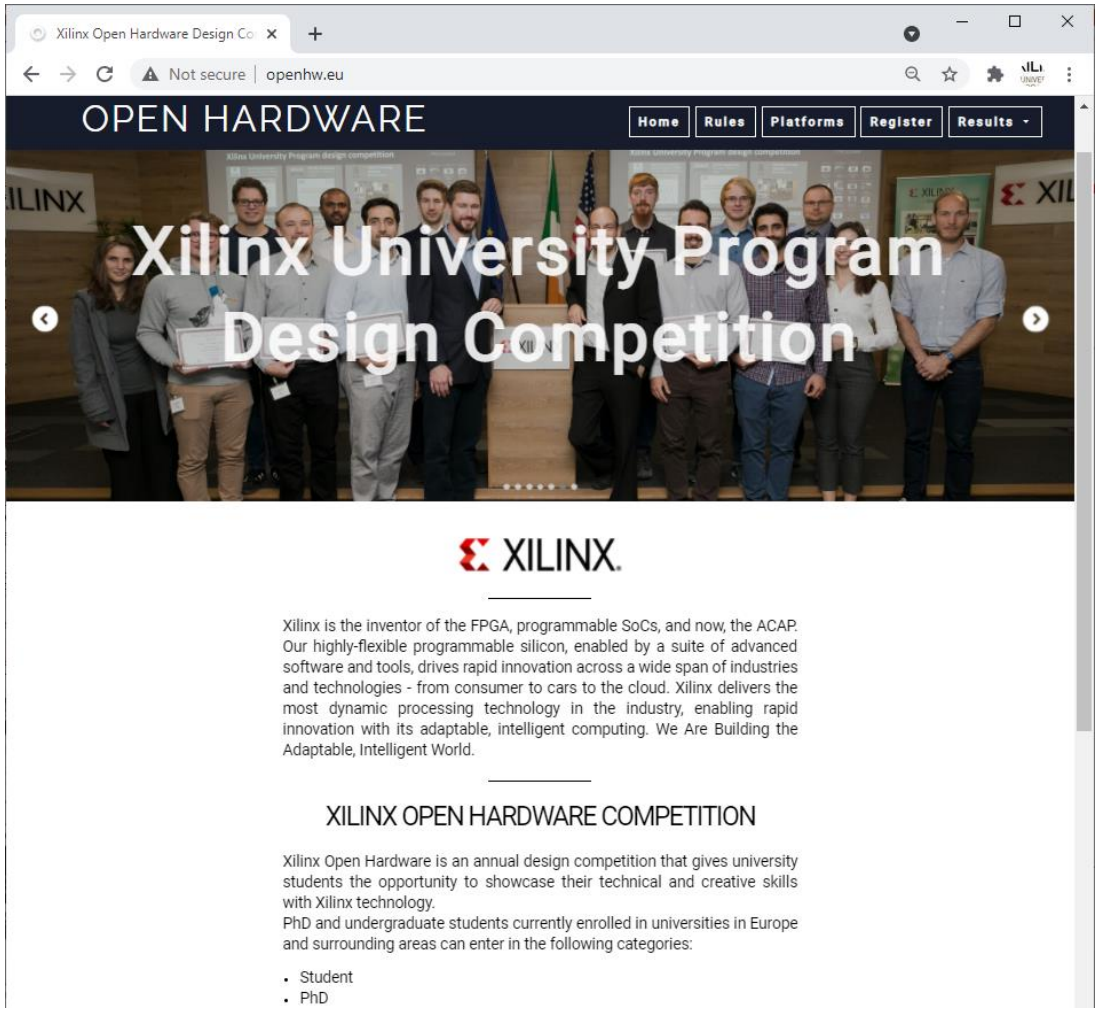
Planned ETH Zurich Expansion

- 10 new nodes with 1 Alveo U55C each to be added
- Opportunity to scale-out applications with homogeneous hardware

Compute Resources	
Look-up Tables (LUTs)	1,304K
Registers	2,607K
DSP Slices	9,024
Dimensions	
Full Height; Half Length; Single Slot	
Memory	
HBM Memory Capacity	16 GB
HBM Total Bandwidth	460 GB/s
Physical Interfaces	
PCI Express	Gen3x 16, 2 x Gen4x 8
Network Interfaces	2x QSFP28

XILINX

Open Hardware 2022 registration now open



- Individuals or teams up to 3
- Cash prizes!
- Categories
  - Student
  - PhD
  - PYNQ
  - Compute Acceleration
- Key dates
  - Register by 28 February 2022
  - Submit by 30 June 2022

www.openhw.eu

Questions?

XILINX.

Thank You

Xilinx Mission

Building the Adaptable, Intelligent World