

# **Enhancing HPC Education and Workflows with Novel Computing Architectures**

Jeffrey Young • Aaron Jezghani • Jeffrey Valdez

Sam Jijina • Xueyang Liu • Michael D. Weiner

Will Powell • Semir Sarajlic

Georgia Institute of Technology



# Adapting to a Changing Compute Landscape

We present a case study of applying traditional HPC techniques to novel architectures, including but not limited to FPGAs, neuromorphic devices, quantum compilers, and SmartNICs.

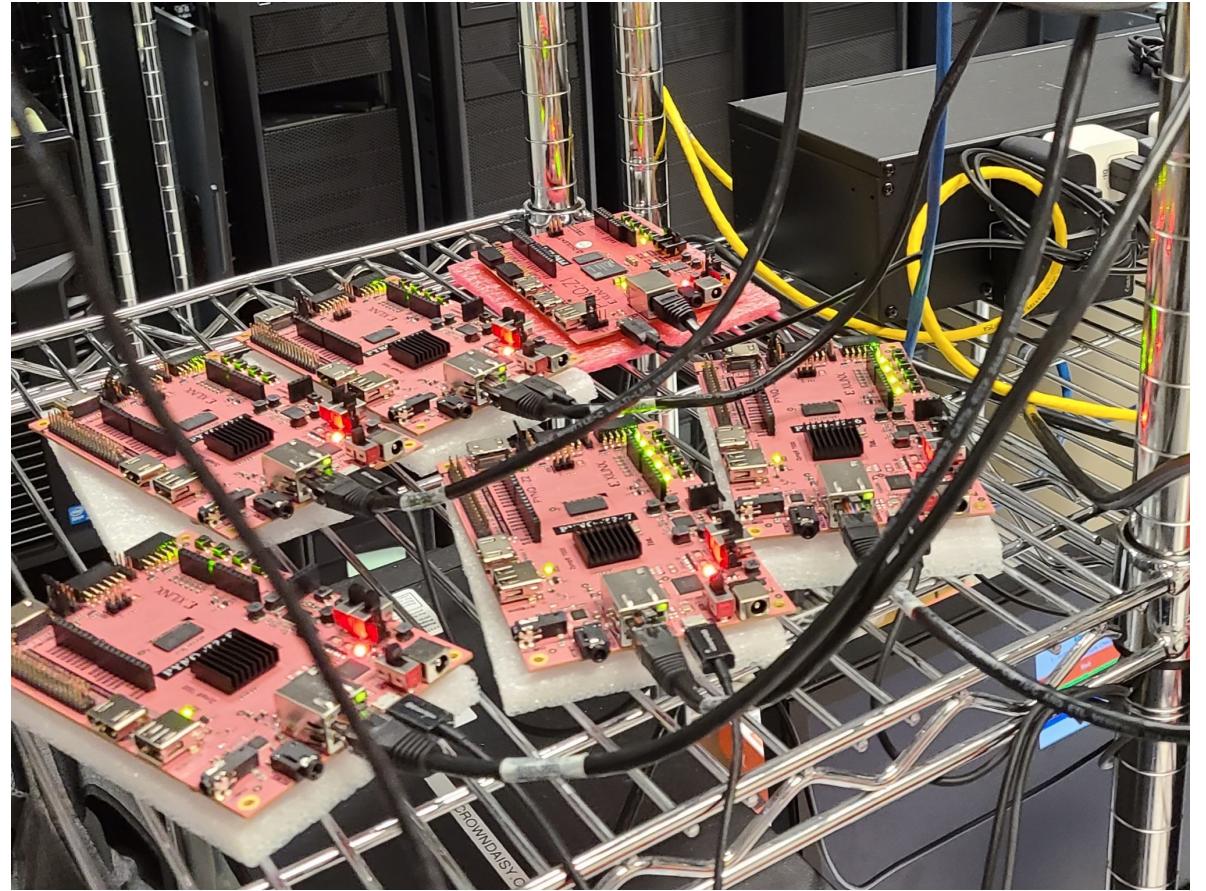
Specifically, we looked at the following questions:

- How can we support larger numbers of students for a ***novel FPGA cluster*** that ties in with an existing HPC system, GT's Instructional Cluster Environment (ICE)?
- How do we support key requirements like ***data separation and privacy for student data*** while allowing for free flow of data between isolated clusters and local access points?
- What types of training would students require to migrate from "hands-on" infrastructure to a remotely scheduled cluster that uses a scheduler like Torque/Moab or Slurm?



# Challenges with Existing Solutions

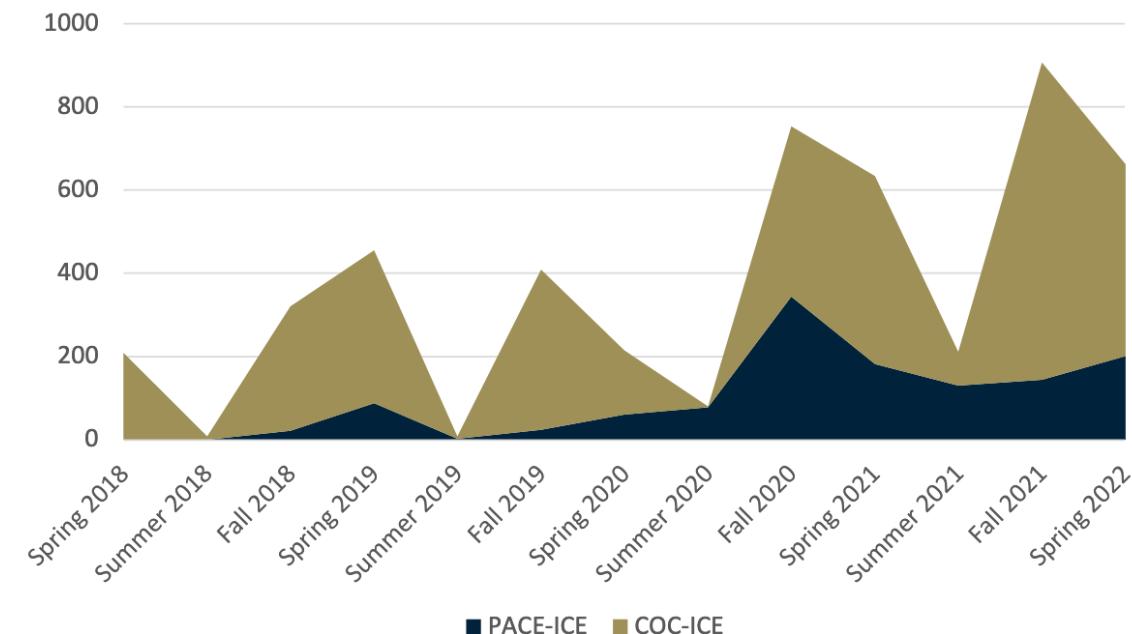
- Two courses with 90+ (CS 3220 and 40+ (ECE 8803) students
- Typical PYNQ FPGA workflows require:
  - The usage of Vivado and Vitis tools with large memory requirements
  - Hands-on interaction with a board to set up Jupyter webserver interfaces
- Supply chain issues limit the number of boards we can deploy



*How can we best support remote FPGA development and usage at scale?*

# Enterprise HPC Expertise to Facilitate Access

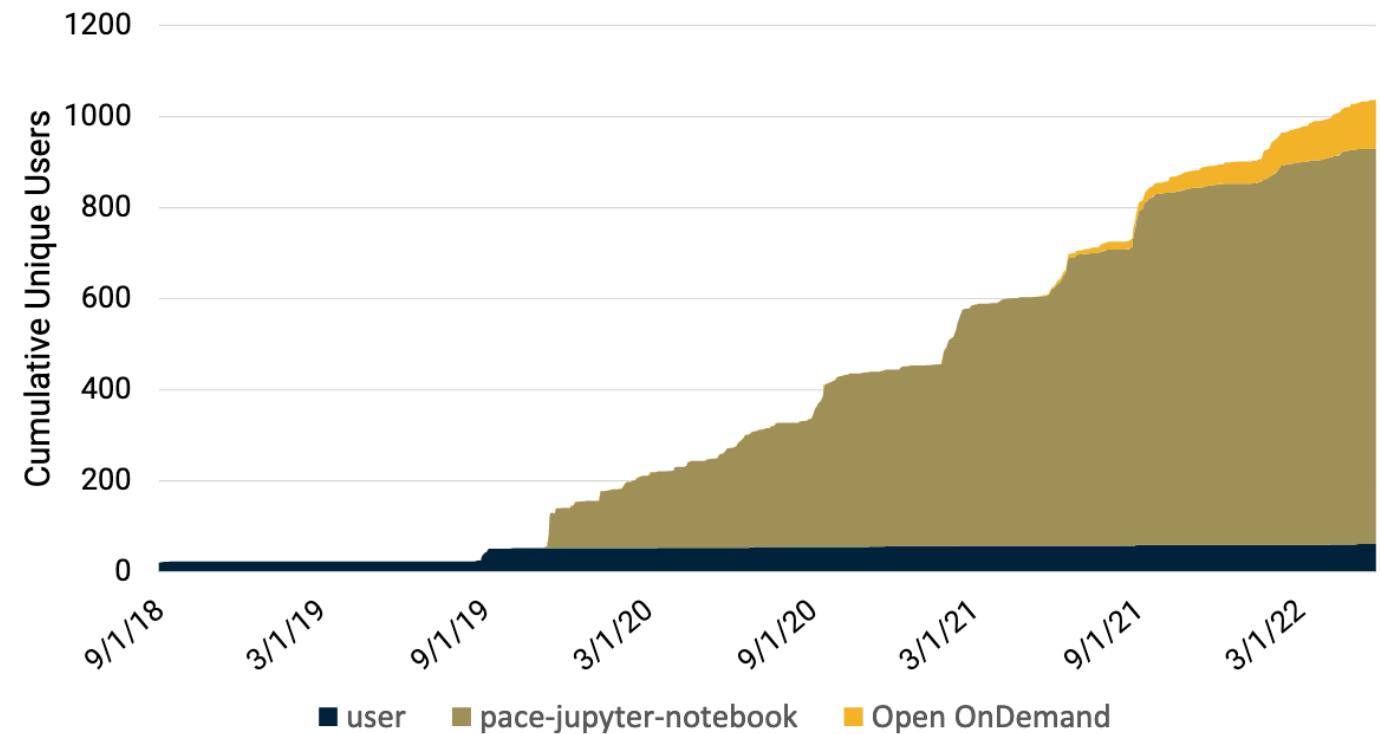
- PACE's Instructional Computing Environment (ICE) and the College of Computing Instructional Computing Environment (CoC-ICE) support over 1600 students and 20 classes in a given year (as of 2021)
  - CoC-ICE has 45 CPU/GPU nodes and is connected via InfiniBand
  - Traditional Torque/Moab scheduler that mirrors larger Phoenix and Hive Systems



# Jupyter Wrapper to Lower Learning Barrier

PACE has deployed custom Torque/Moab Jupyter and VNC scripts that support notebook and GUI-based app usage

- Since its introduction in 2019, these efforts have dramatically increased utilization and the overall userbase

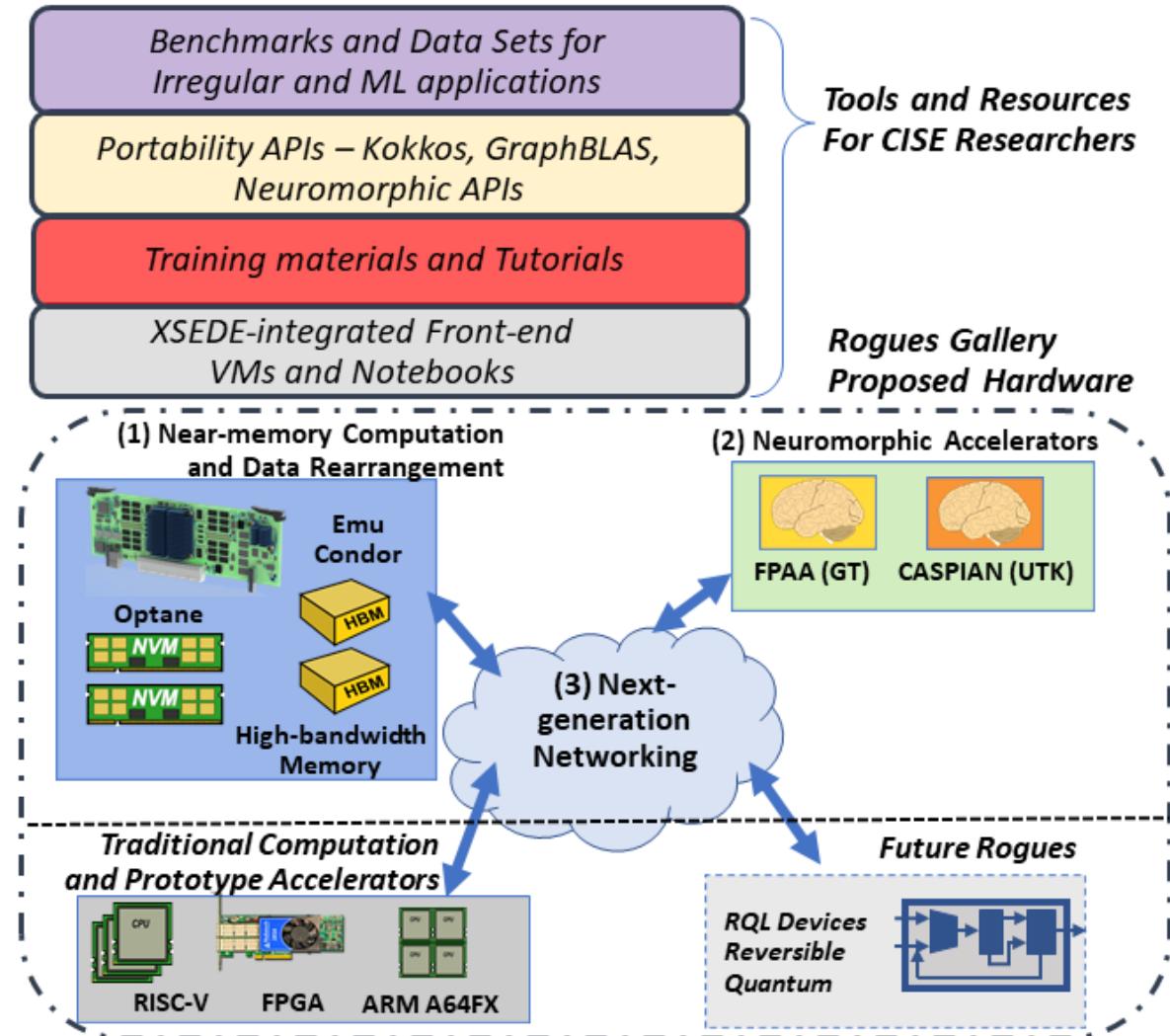


# CRNCH Rogues Gallery

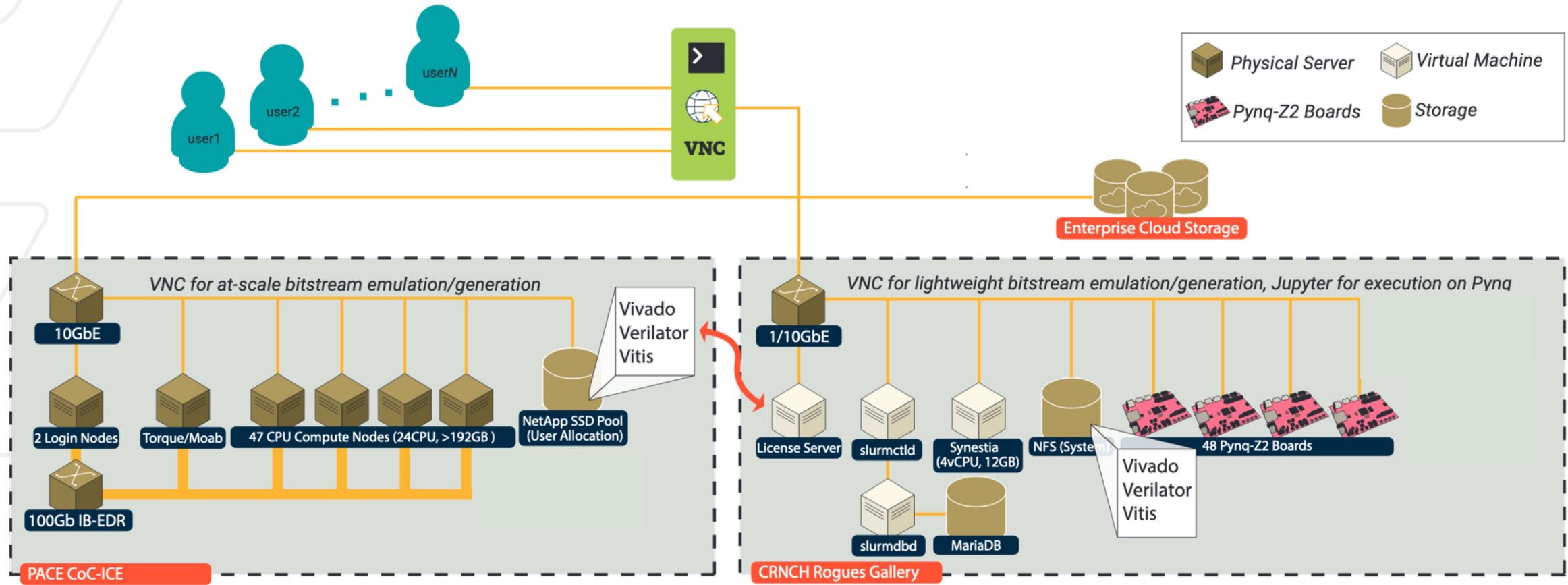
The Rogues Gallery is an NSF funded testbed that is focused on increasing access to novel architectures for CISE researchers.

Critically, these systems must be **accessible** with appropriate training materials.

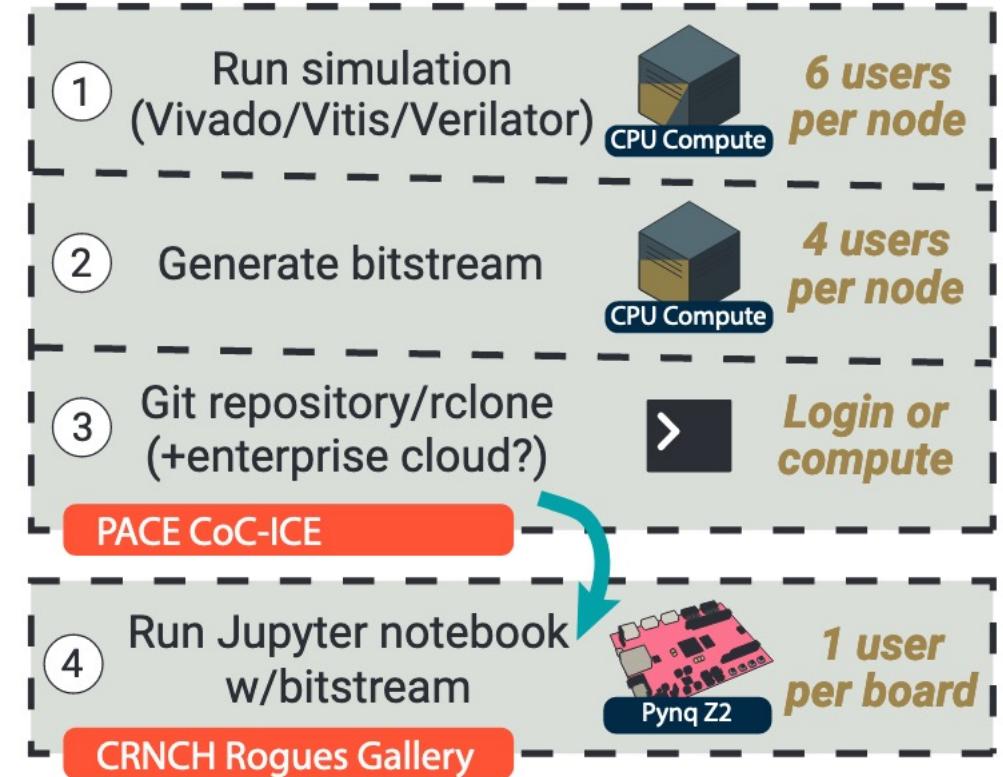
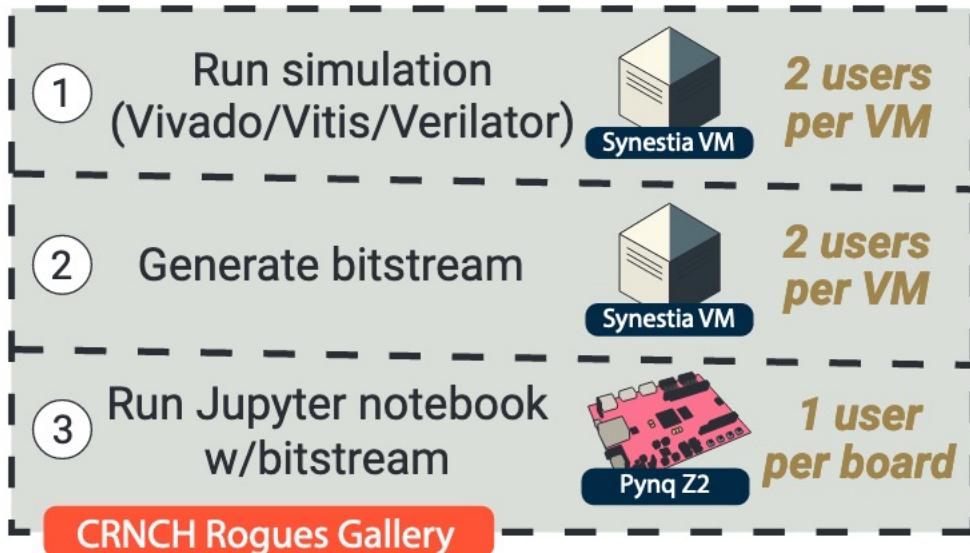
- Georgia Tech courses provide good motivation to enhance these features of the testbed!



# CRNCH + PACE PYNQ Infrastructure



# Two User Workflows



A combined workflow allows for a dramatic reduction in the bottleneck of design and initial testing

- However, there are some issues to consider related to data security

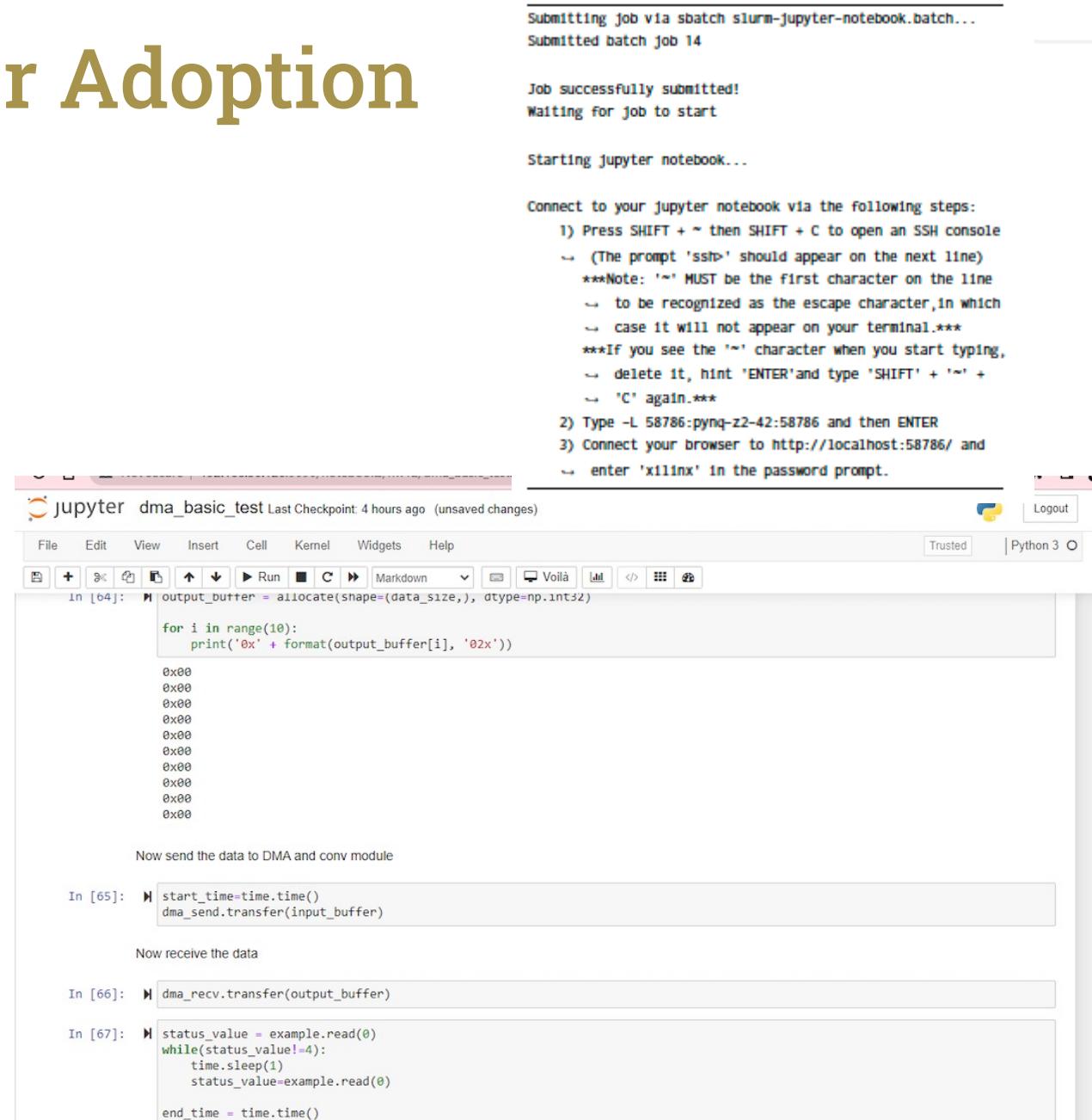
# Spring 2022 PYNQ Cluster Adoption

42 PYNQ Z2 FPGAs were deployed along with Vitis/Vivado 2021.1 tools on CoC-ICE and CRNCH-hosted VMs.

CoC-ICE was used to simulate code and generate bitstreams, and students used GT GitHub or rclone to migrate files between ICE and CRNCH RG testbed

- CRNCH nethome was set up with a class-specific folder login that could be wiped at the end of the semester.

Students used SSH port forwarding and Slurm to request an interactive PYNQ Jupyter session.



The screenshot shows a Jupyter Notebook titled "jupyter dma\_basic\_test". The notebook has a "Last Checkpoint: 4 hours ago (unsaved changes)" message. The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and various icons. A sidebar on the right shows "Trusted" and "Python 3". The code cell In [64] contains:

```
for i in range(10):
    print('0x' + format(output_buffer[i], '02x'))
```

The output shows the first 10 elements of the buffer as zeros:

```
0x00
```

The text "Now send the data to DMA and conv module" is displayed below the code cell. The next code cell In [65] contains:

```
start_time=time.time()
dma_send.transfer(input_buffer)
```

The text "Now receive the data" is displayed below the code cell. The next code cell In [66] contains:

```
dma_recv.transfer(output_buffer)
```

The final code cell In [67] contains:

```
status_value = example.read(0)
while(status_value!=4):
    time.sleep(1)
    status_value=example.read(0)

end_time = time.time()
```

At the top of the slide, there is a status bar with the following text:

```
Submitting job via sbatch slurm-jupyter-notebook.batch...
Submitted batch job 14

Job successfully submitted!
Waiting for job to start

Starting jupyter notebook...
```

Below the status bar, instructions for connecting to the Jupyter notebook are provided:

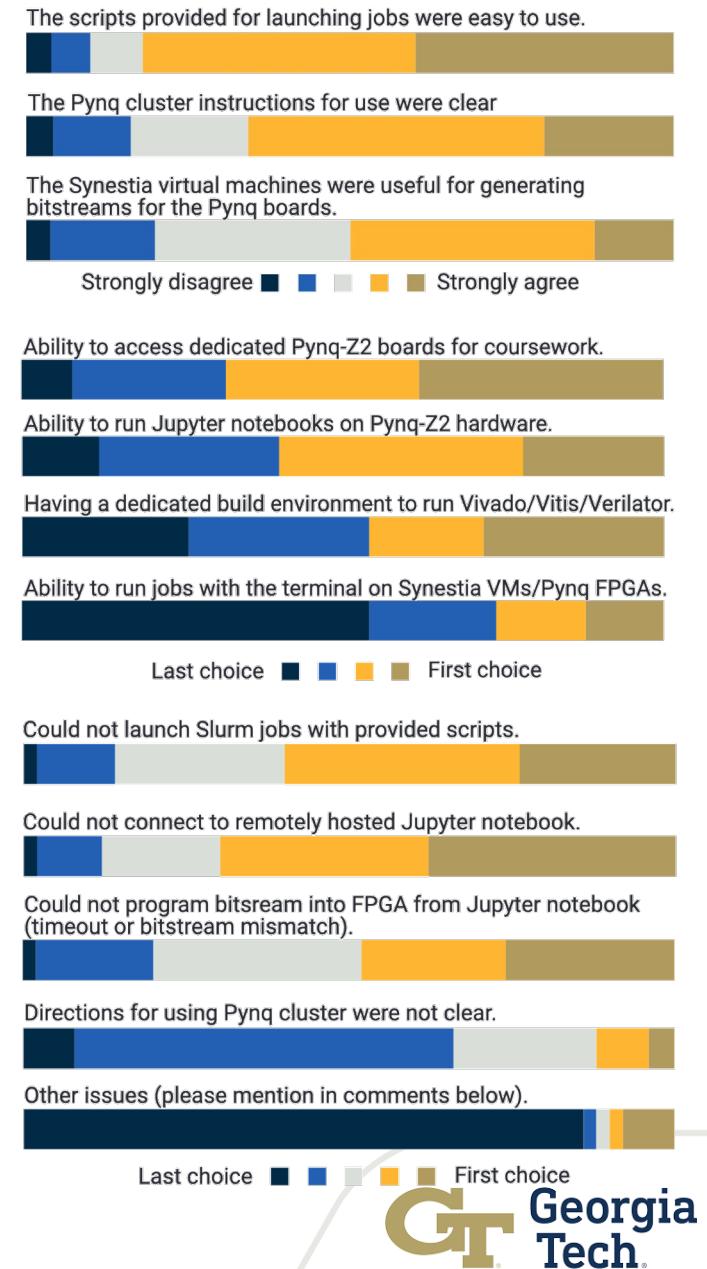
Connect to your jupyter notebook via the following steps:

- 1) Press SHIFT + ~ then SHIFT + C to open an SSH console
  - ↳ (The prompt 'sshr' should appear on the next line)
  - \*\*\*Note: '~' MUST be the first character on the line
  - ↳ to be recognized as the escape character, in which case it will not appear on your terminal.\*\*\*
  - \*\*\*If you see the '~' character when you start typing, delete it, hit 'ENTER' and type 'SHIFT' + '~' + 'C' again.\*\*\*
- 2) Type -L 58786:pynq-z2-42:58786 and then ENTER
- 3) Connect your browser to <http://localhost:58786/> and enter 'xilinx' in the password prompt.

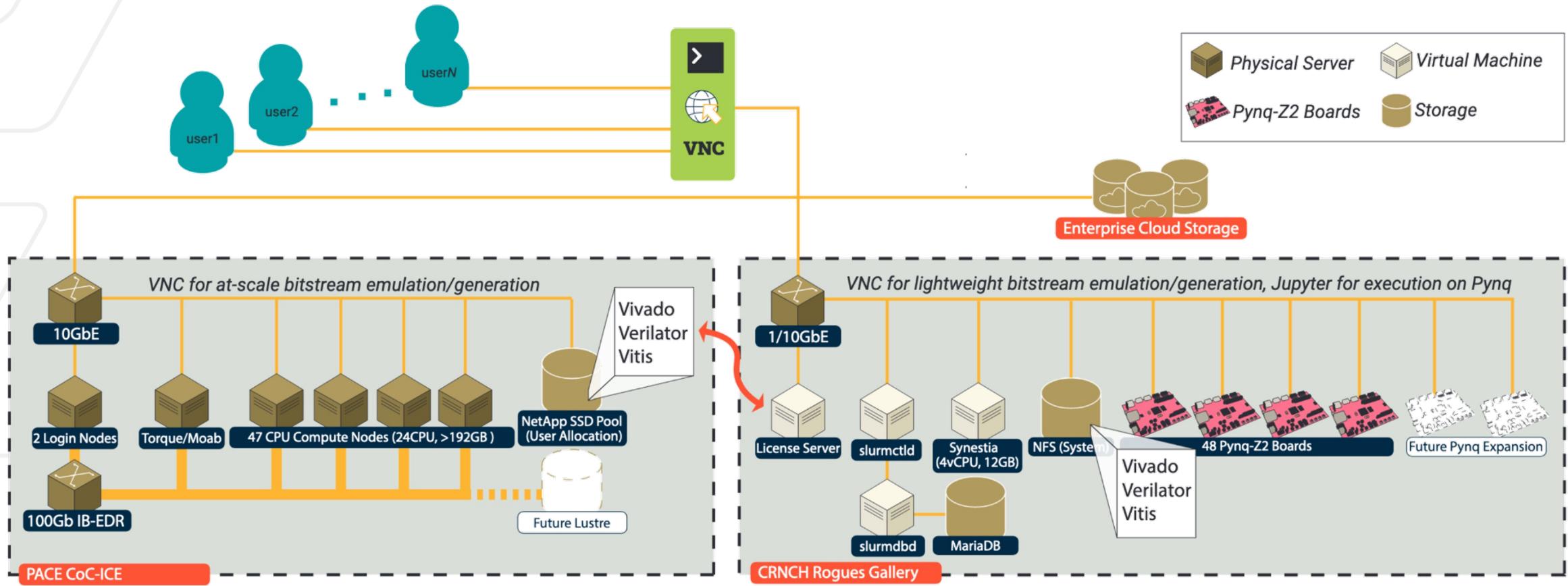
# End of Semester Student Feedback

Students were surveyed post-semester to garner user insights into the infrastructure/workflow efficacy

- Non-anonymous results were given extra credit
- **Pros:** Students appreciated the workflow including using HPC servers for large-memory jobs and the Jupyter interface and scripts to use PYNQ boards. Cost savings from not needing to buy a PYNQ board also exceeds \$8,000 per semester.
- **Cons:** Students encountered errors related to launching Slurm jobs, user contention, and a few issues with PYNQ commands. Lack of Slurm accounting created high-utilization scenarios.



# Cluster Expansions in Response to Feedback



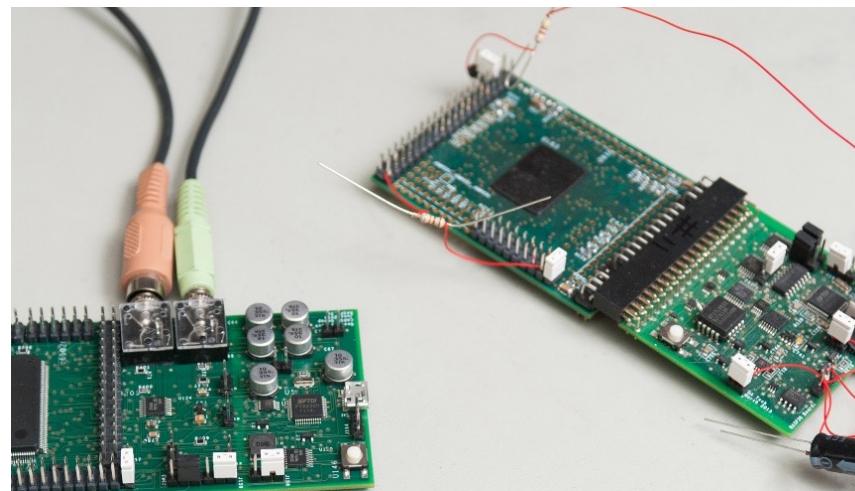
# Future Improvements

We are currently working on fully supporting Open OnDemand, Slurm accounting, and XDMoD reporting

- This would minimize the need for custom scripts to launch jobs
- Accounting would be useful to set better job limits for users for exams, etc.

Other novel devices can follow this same model

- If it can be connected to a Raspberry Pi that can run slurmd, we can schedule it and create a notebook interface!



## CRNCH

CENTER FOR RESEARCH INTO  
NOVEL COMPUTING HIERARCHIES

## ROGUES GALLERY



OnDemand provides an integrated, single access point for all of your HPC resources.

### Message of the Day

This is the RG OnDemand Server. From this interface you can submit jobs to the following:

- rg-emu-dev: VM for compiling/simulating Emu Chick code
- karrawingi-login: login node for the Emu Chick
- rg-fpga-dev-<1-3>: VMs for FPGA compilation with Intel or Xilinx tools
- flubber<1-3>: Servers with FPGAs and small TPUs
- brainard: Desktop connected to Zynq and Pi devices
- rg-fpaa-host: A Raspberry Pi that is connected to our FPAA prototype.
- octavius-login: Login node for the 16-node Arm A64FX cluster

Most tools can be found under the /tools/ netshare folder. For more information on specific systems please see the [wiki](#).

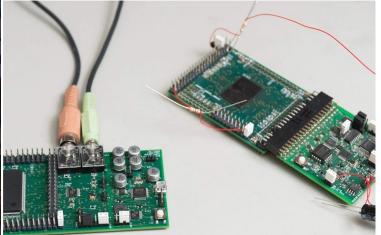
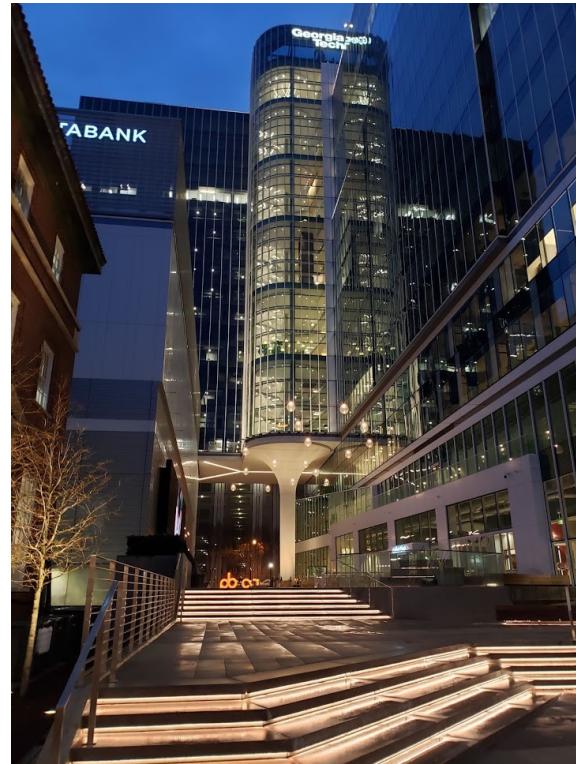
powered by  
 OnDemand

# Conclusion

As post-Moore technologies continue to become mainstream, our courses and HPC resources will need to adapt:

- Novel architectures typically follow a defined simulation/hardware execution workflow that maps well to traditional HPC resources.
- Coursework requires special care for management of student data but existing techniques and tools can be used to bridge novel testbeds and traditional clusters.
- Students may need extra help to learn new concepts like schedulers and SSH port forwarding, but documentation and engagement can help to bridge their transition to remote novel architecture testbeds.

This work is supported in part by “CCRI: Medium: Rogues Gallery: A Community Research Infrastructure for Post-Moore Computing” (NSF CNS-2016701) and “MRI: Acquisition of an HPC System for Data-Driven Discovery in Computational Astrophysics, Biology, Chemistry, and Materials Science” (NSF OAC-1828187).



Learn more about CRNCH RG and request an account at <https://crnch-rg.cc.gatech.edu/> and PACE at <https://pace.gatech.edu>