Onboarding Users to A64FX via Open OnDemand

A. Jezghani • K. Manalo • W. Powell J. • Valdez • J. Young
Georgia Institute of Technology, Atlanta, GA

IWAHPCE-2022
International Workshop on Arm-based HPC: Practice and Experience

Georgia Tech

A Brief Glimpse of Research Computing at GT…

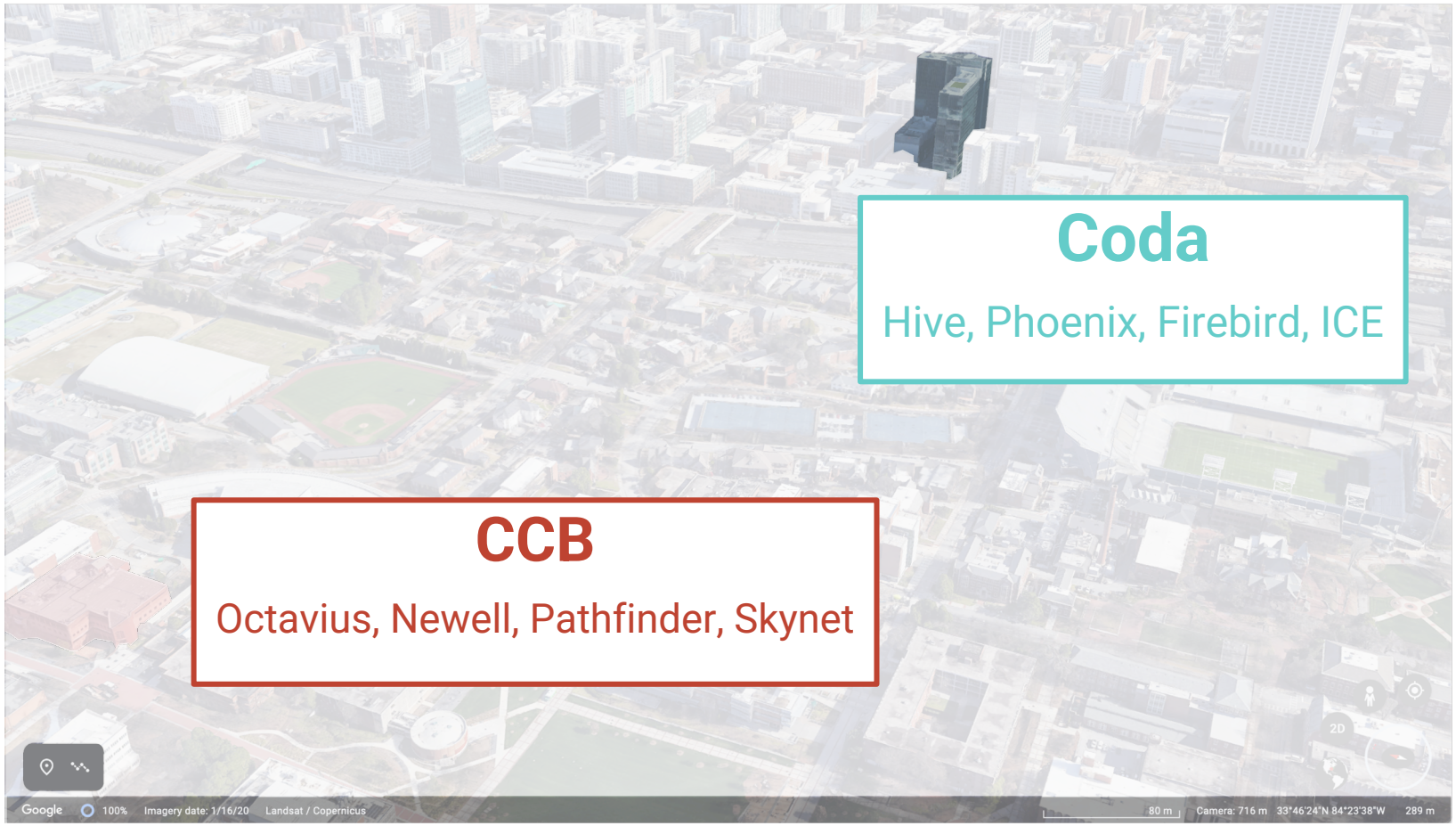
Aaron Jezghani
PACE

Kevin Manalo
PACE

William Powell
CSE

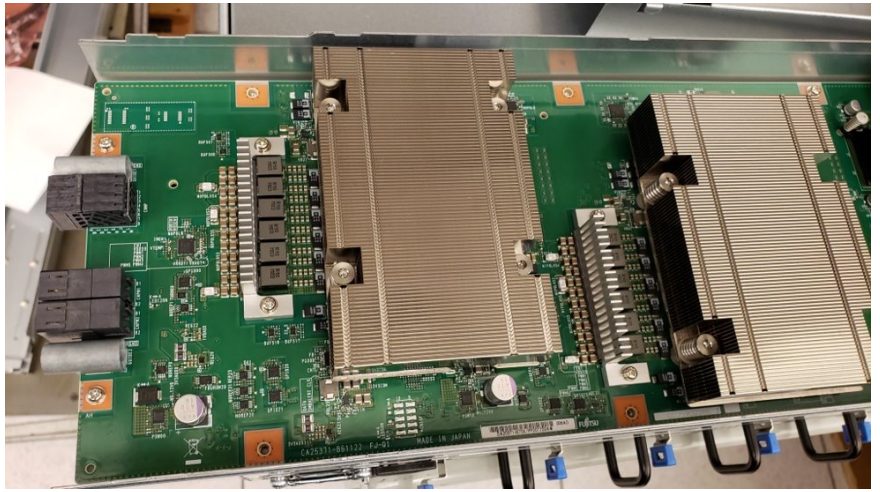
Jeffrey Valdez
IC

Jeffrey Young
CS

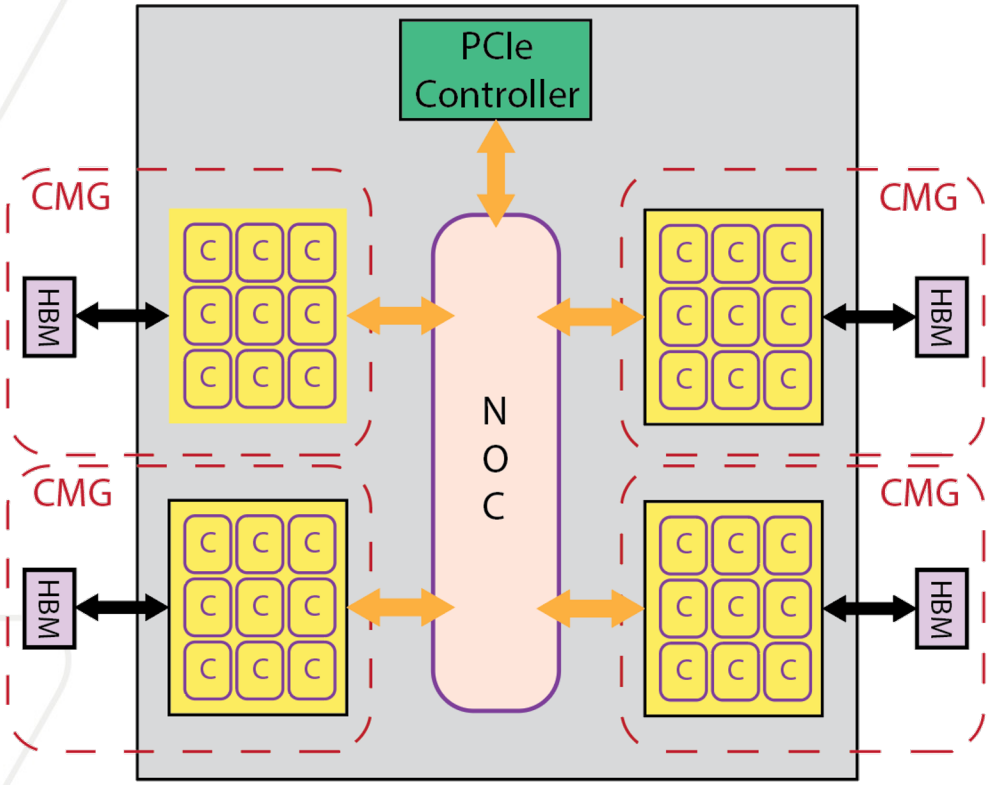


Octavius: Hive's Rogue Little Brother

- Octavius is supported by the National Science Foundation
  - 16 A64FX nodes are part of NSF MRI award #1828187 to allow exploration of Arm+SVE as an alternative to AVX512
  - Hosted within the Center for Research into Novel Computing Hierarchies (CRNCH) Rogues Gallery testbed, funded by NSR award #2016701
- Timeframe and architecture mirror Stonybrook's Ookami (but at a smaller scale)
  - Deployed in Fall 2020
  - 2x HPE Apollo80 chassis, each with 8 nodes



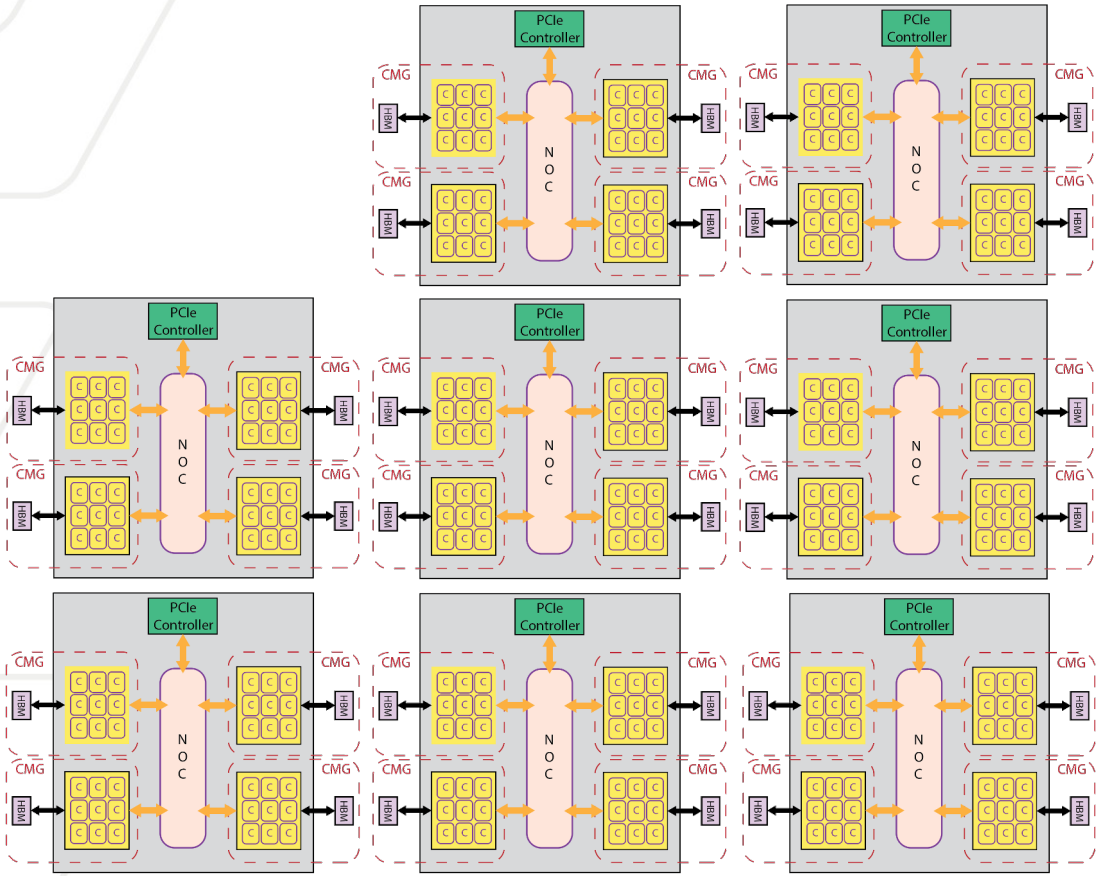
Octavius: NUMA (and Name)



- >2.7 TFLOPS
- Supports Scalable Vector Extension (SVE)
  - Enables Vector Length Agnostic (VLA) programming, which contributes to portability, scalability, and optimization
- 4 Core Memory Groups (CMGs)
  - 12 cores
  - 64KB L1$ per core
    - 256b cache line
  - 8MB L2 shared between all cores
    - 256b cache line
  - 0 L3$
  - 8GB HBM at 256GB/s

http://www.jicfus.jp/jp/wp-content/uploads/2018/11/msato-190109.pdf

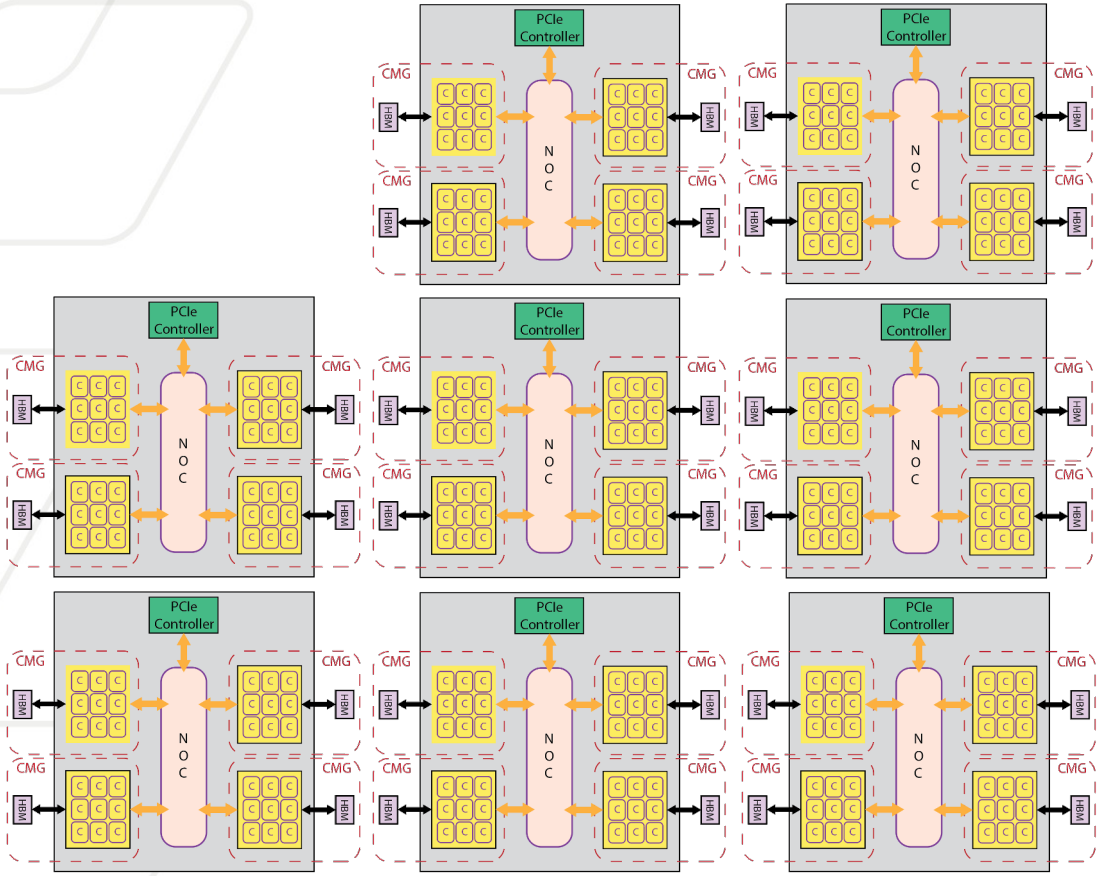
Octavius: NUMA (and Name)



?
=

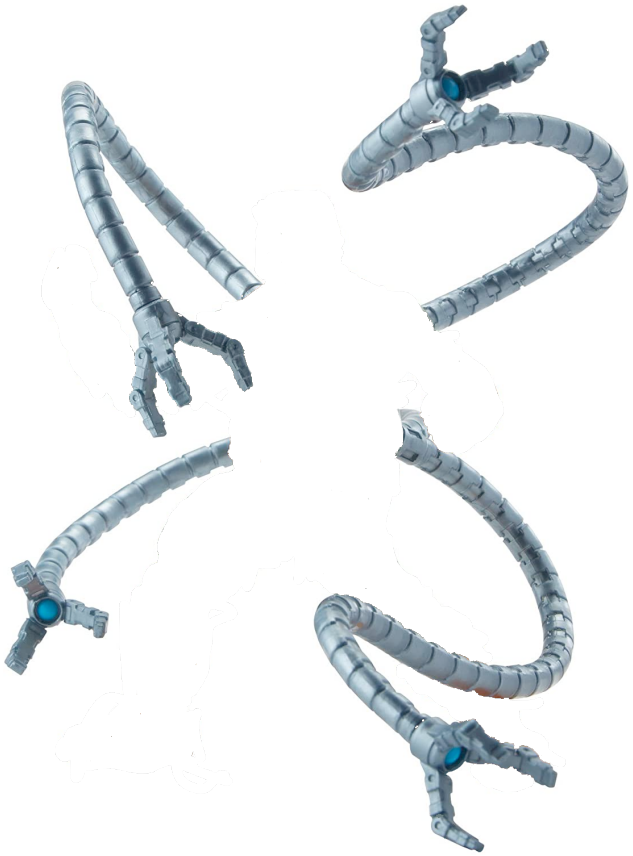
http://www.jicfus.jp/jp/wp-content/uploads/2018/11/msato-190109.pdf

Octavius: NUMA (and Name)



!

=



Georgia Tech

Octavius: Environment and Software

- x86 VM login node
- Slurm 21.08 manages Octavius access (and other RG compute)
  - Debug queue to test/compile code before running at scale
- Cray, Arm, and Gnu compiler/library modules
  - Expect insights to also be applicable to larger clusters like Ookami
- Shared home and NVMe-backed scratch storage

The Hurdles of New Technologies and Users

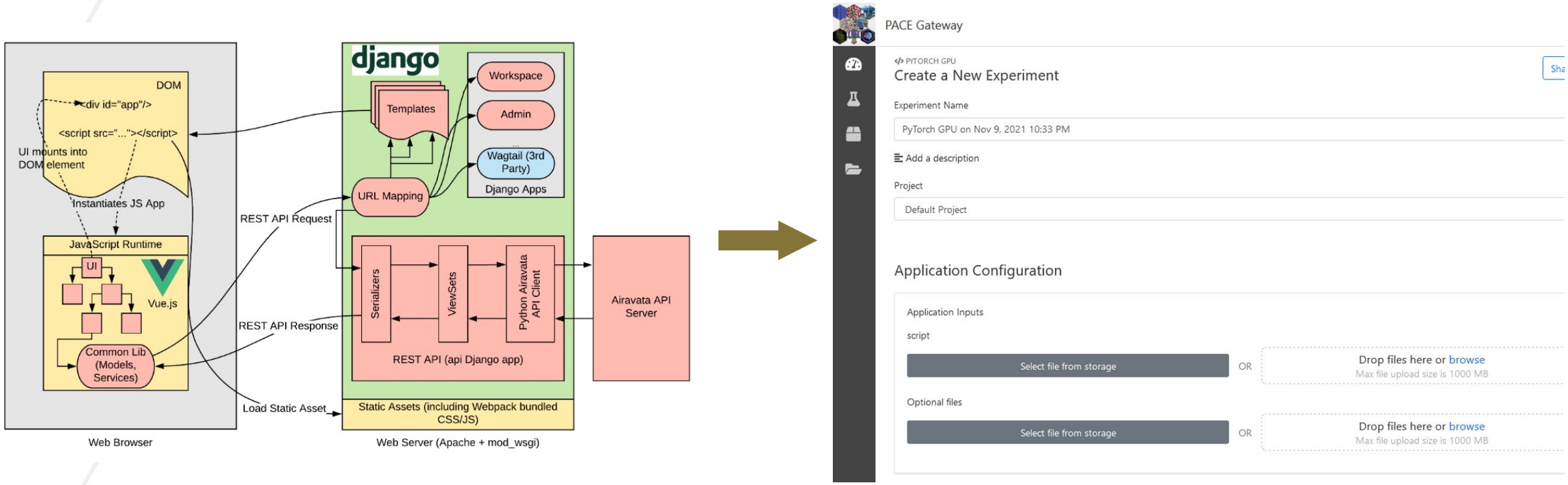
- As exciting as algorithm exploration and software porting might be for veteran users…
  - The “tried-and-true” tools often take several cycles to adapt
    - This happens with updates to traditional architectures too
    - Vendors contribute heavily to upstream, but it can still take time to trickle down
  - Vendor-provided tools are great to use!
    - …but even as a port of something familiar, there’s still a learning curve
    - And these tools may be proprietary!
- Meanwhile, new users are constantly inbound
  - Remote command line access can be intimidating (and feel prohibitive)
  - Even for experienced users, interactive server workflows can be problematic

These Aren't New Issues! But...

- Knowledge transfer can be a difficult process
  - Documentation relies on the "Goldilocks principle"...on a user-by-user basis.
  - Trainings are great but might not be convenient for the users.
  - Command-line self-service tutorials may not include the instructive element available in other forms.
- Many solutions are less than elegant
  - Performance problems (X11 forwarding)
  - Scalability issues (FastX licensing, X2Go resource needs)
  - Less-than intuitive interfaces (SSH tunneling)
  - Client application requirements (e.g. VNC)
  - Local security rules can be an impediment (Port blocking)

Gateways Expand the Impact of Clusters…

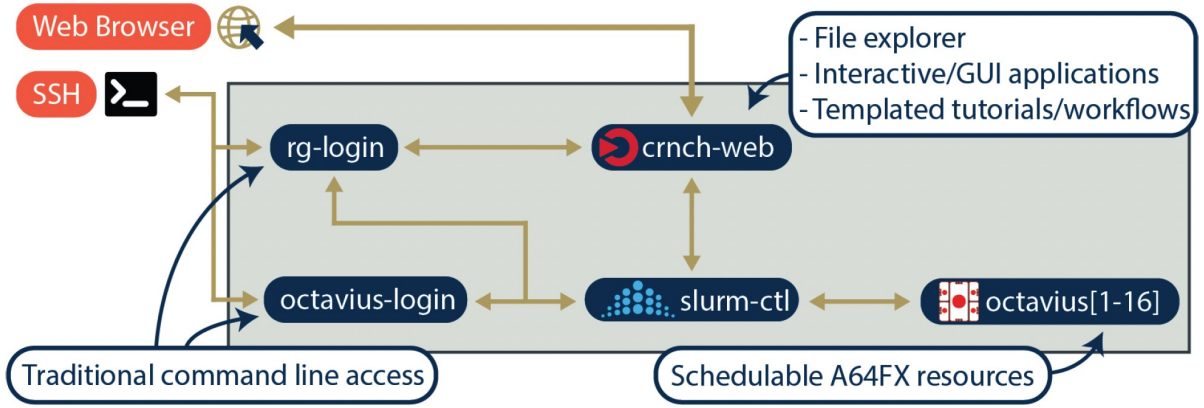
For example, Software as a Service via Django + Apache Airavata:



https://dl.acm.org/doi/pdf/10.1145/3311790.3396650

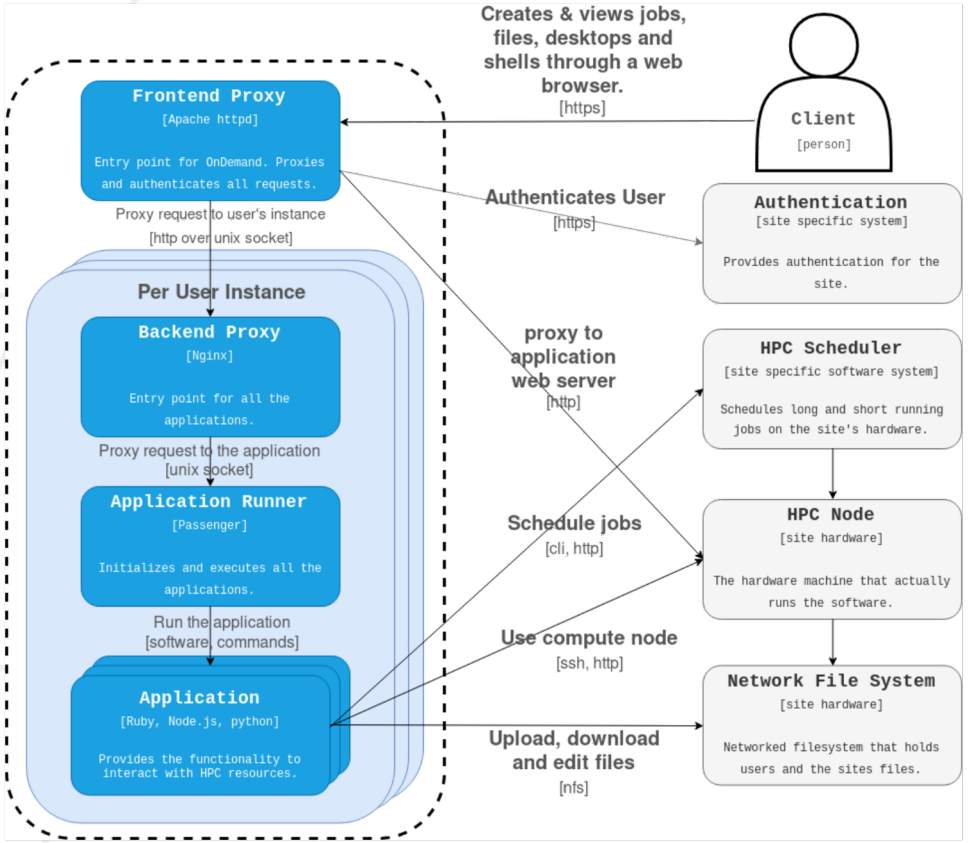
…But Emerging Technologies Demand Flexibility

- Before software can be available as a service, the software needs to exist
  - Optimizing existing software for new architectures is a time-intensive process
  - New architectures can tackle new problems, which may or may not be identified
- Enter Open OnDemand (OOD)
  - Developed and maintained by OSC
  - Complements traditional CLI workflow
  - Reduced time to compute



Hudak, D., et al., 2018. Open OnDemand: a web-based client portal for HPC centers. *Journal of Open Source Software*, *3*(25), p.622.

How Does OOD Work?



- Users connect to the Apache frontend
  - Shibboleth, CAS, OpenID, SAML, PAM, etc. supported for authentication
- Backend Nginx spawned per user (PUN) to interface with the cluster
  - Proxy request to various passenger applications, e.g. execute an interactive compute job or engage the filesystem
- **FULL HPC ACCESS VIA BROWSER**

Installing and Configuring OOD

- Most of the heavy lifting is on the OOD server (x86) itself, which can be a basic
  - Available as RPMs for CentOS/RHEL 7/8 (this is what we did)
  - Documentation fully details installation
    - Manual build of mod_auth_cas on CentOS/RHEL 8, not included in documentation
  - 3 main YAML configuration files
    - /etc/ood/config/ood_portal.yml
    - /etc/ood/config/nginx_stage.yml
    - /etc/ood/config/clusters.d/<cluster>.yml
  - *ood-portal-generator* utility to update actual web server configurations with OOD default values and custom definitions from the YAML files

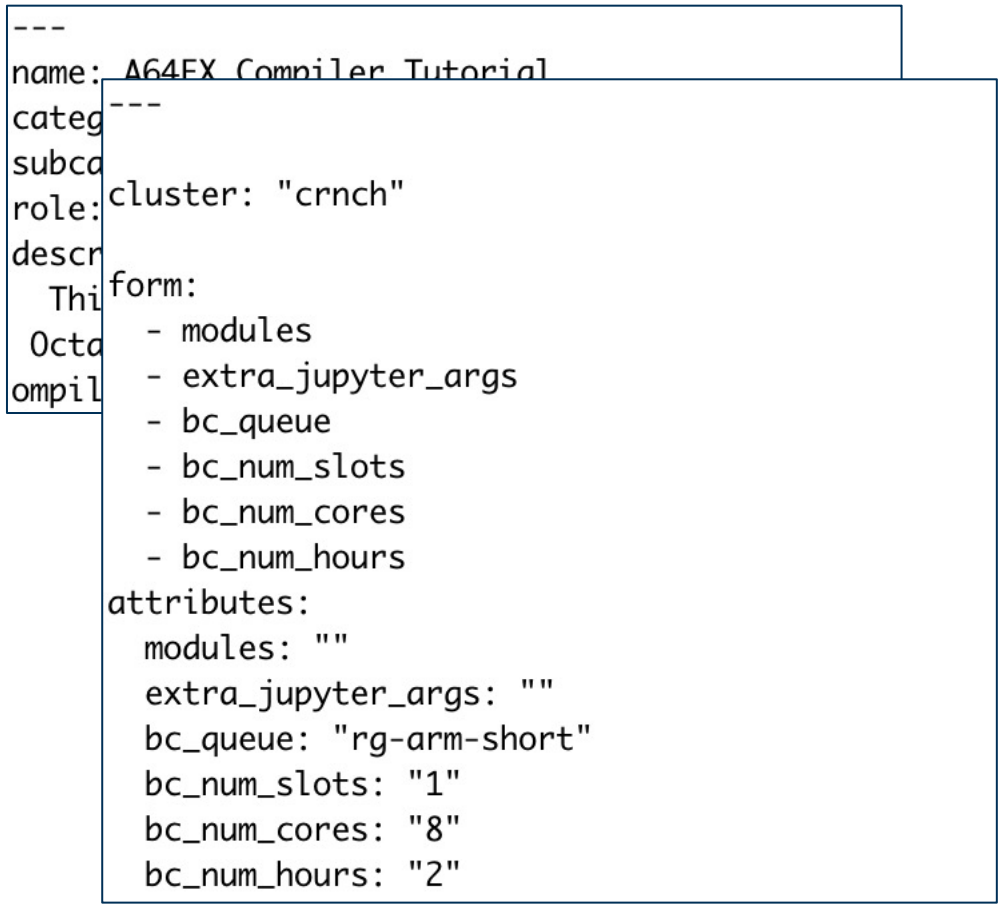
Installing and Configuring OOD (cont'd)

- Passenger applications written as a combination of YAML configurations, embedded ruby, javascript, and python. Each application includes:
  - manifest.yml

```
---
name: A64FX Compiler Tutorial
category: A64FX
subcategory: Tutorials
role: batch_connect
description: |
  This app will launch a Jupyter Notebook on an
 Octavius A64FX node to explore the available c
ompiler and performance library stack.
```

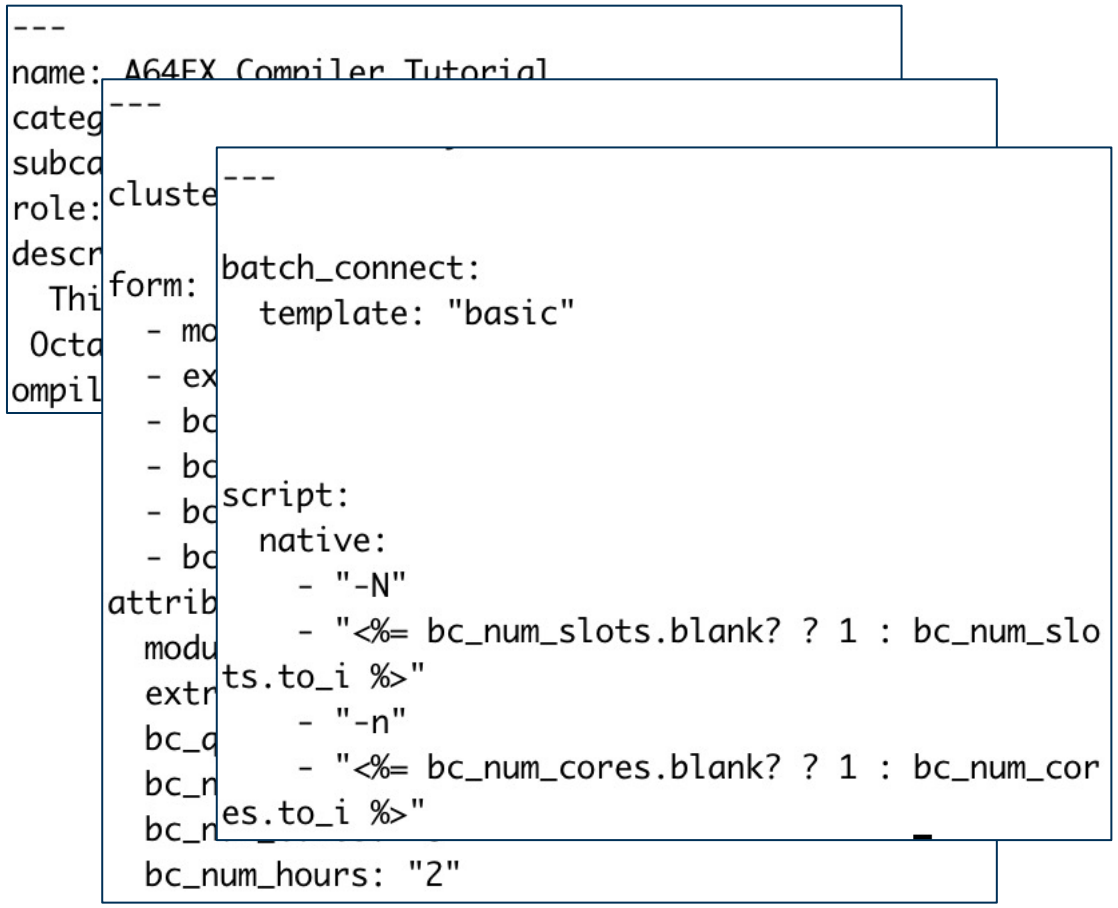
Installing and Configuring OOD (cont'd)

- Passenger applications written as a combination of YAML configurations, embedded ruby, javascript, and python. Each application includes:
  - manifest.yml
  - form.yml(.erb)



Installing and Configuring OOD (cont'd)

- Passenger applications written as a combination of YAML configurations, embedded ruby, javascript, and python. Each application includes:
  - manifest.yml
  - form.yml(.erb)
  - submit.yml(.erb)
- Passive user workflow orchestration
  - Guide them to the right resources per workflow/hardware/policy



Georgia Tech

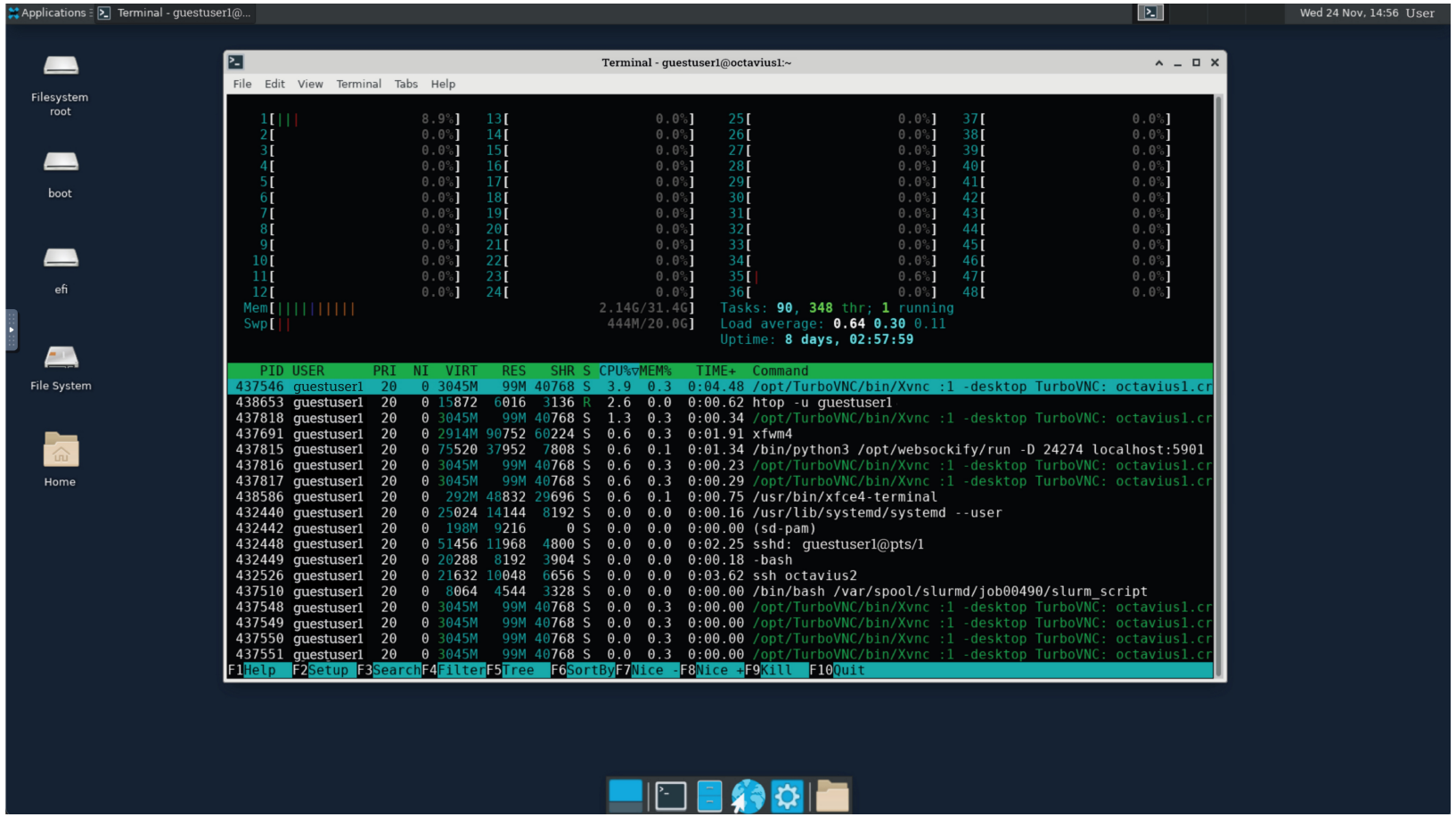
Configuring Compute Nodes for OOD

- Most OSC apps that leverage VNC expect xfce + TurboVNC
  - Aarch64 xfce is available from EPEL, but libturbojpeg needs to be manually built with the appropriate flag
- Additionally, need websockify
  - Easily installed via system pip
- Beyond these basic dependencies, any software called by the apps needs to be present on the compute node
  - Each application is customized with an associated batch script to load modules, set environment, feed content through socket, etc.

Scaling Concerns?

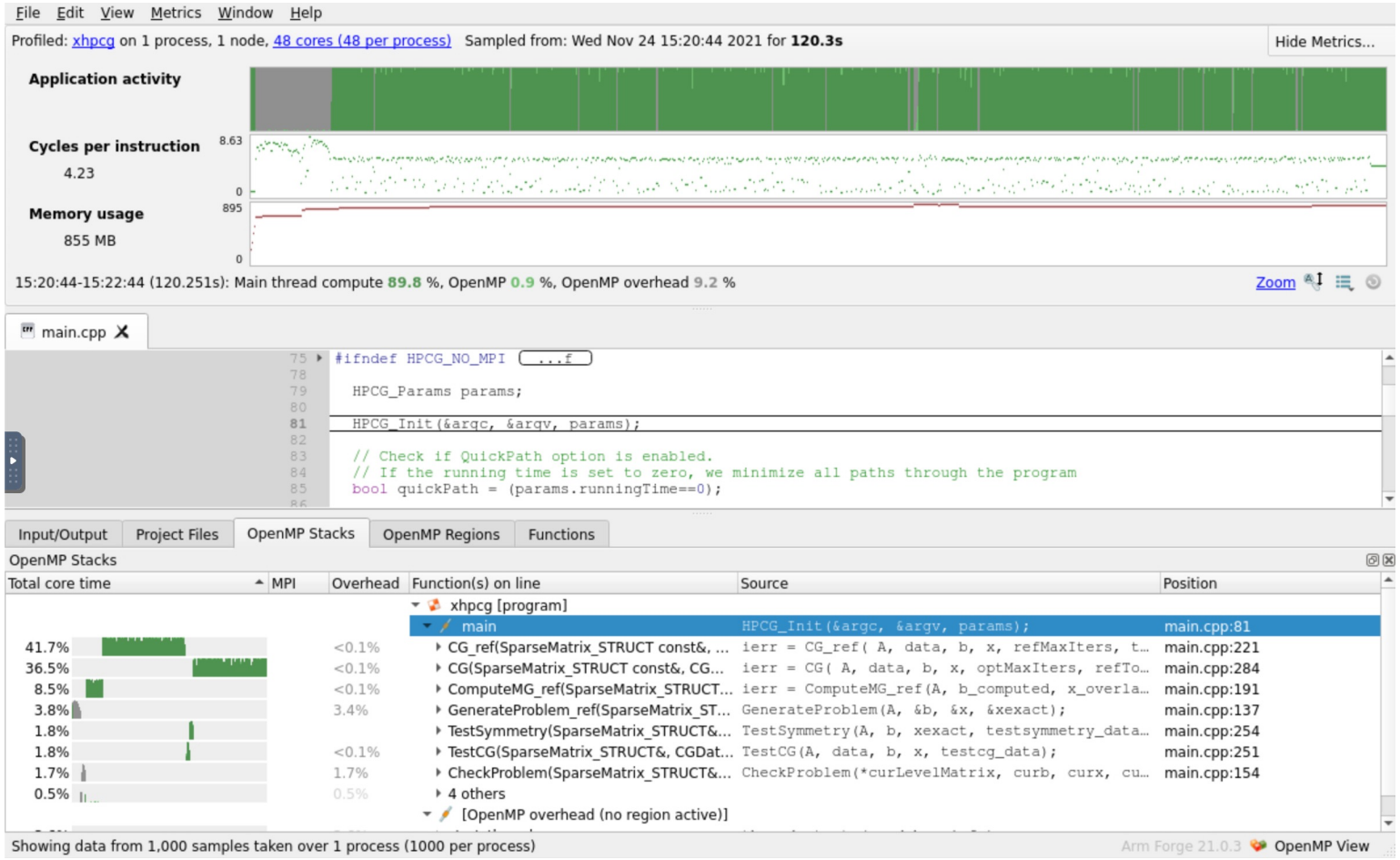
- Applications spawn on compute nodes, but PUN does have overhead
  - The need to reconnect to existing PUNs introduces some challenges for traditional load balancing
  - OSC reports supporting 600 unique users/month, 60 concurrent
  - PACE recently explored OOD VM capacity with WebLOAD
    - Initial tests of 12c/12GB VM indicated max ~50 concurrent users
    - At 16c/64GB VM, performance appeared stable in excess of 100 users

Virtual Desktop



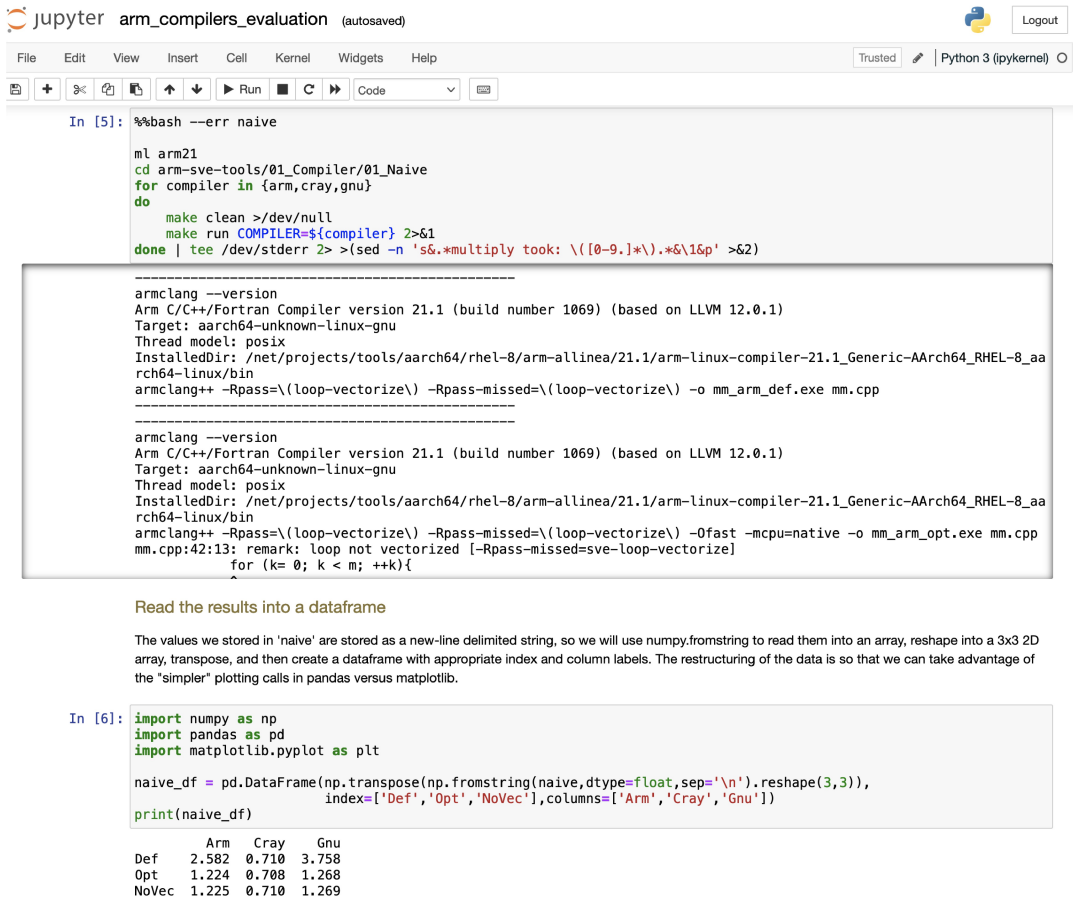
Georgia Tech

Arm Forge Suite



Georgia Tech

Arm Compiler Tutorial



Conclusion

- Onboarding new users can be a challenge
- OOD is a powerful platform to lower the barrier to entry
  - Provides full cluster access while improving GUI/interactive experiences
- Can leverage applications and example code to produce self-service tutorials
- Similarly, can leverage the dynamic content capabilities to abstract complex workflows (such as containerized solutions)
- Expanding Rogues Gallery OOD capabilities and looking to grow the community
  - Encourage innovative contributions to A64FX using Octavius

Acknowledgements

*This research was supported by the **NSF MRI award #1828187**: "MRI: Acquisition of an HPC System for Data-Driven Discovery in Computational Astrophysics, Biology, Chemistry, and Materials Science." Additionally, this research was supported in part through research infrastructure and services provided by the Rogues Gallery testbed hosted by the Center for Research into Novel Computing Hierarchies (CRNCH) at Georgia Tech. The Rogues Gallery test bed is primarily supported by the National Science Foundation (NSF) under **NSF Award Number #2016701**. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s), and do not necessarily reflect those of the NSF.*

Georgia Tech