Onboarding Users to A64FX via Open OnDemand

Aaron Jezghani
ajezghani3@gatech.edu
OIT-PACE
Georgia Institute of Technology
Atlanta, Georgia, USA

Kevin Manalo
OIT-PACE
Georgia Institute of Technology
Atlanta, USA

William Powell
CSE
Georgia Institute of Technology
Atlanta, USA

Jeffrey Valdez
IC
Georgia Institute of Technology
Atlanta, USA

Jeffrey Young
SCS
Georgia Institute of Technology
Atlanta, USA

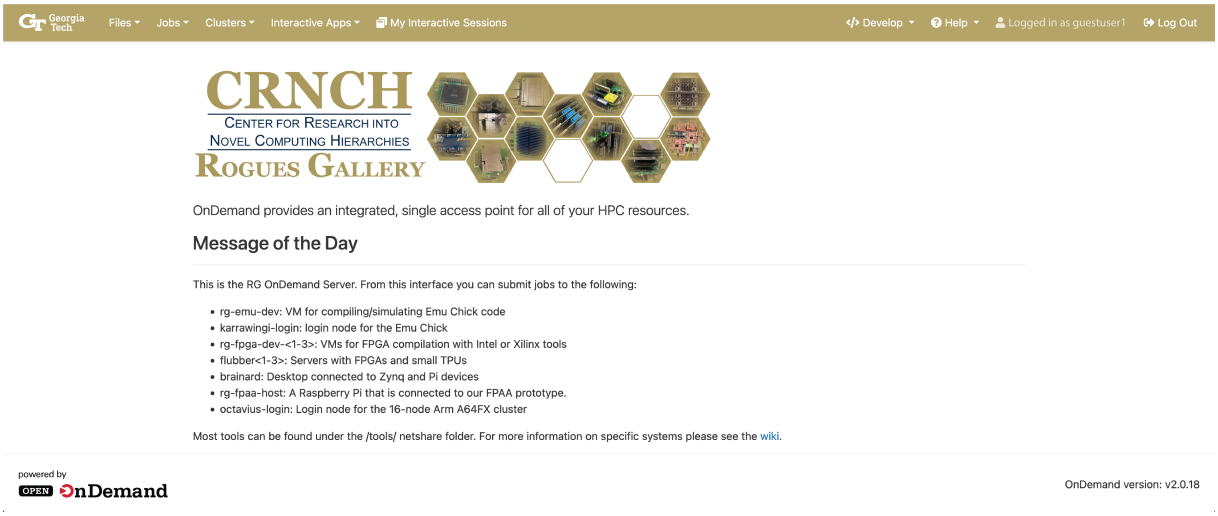

Figure 1: Octavius testbed infrastructure and access, including Open OnDemand. This browser-based interface seamlessly provides scheduled access to the same A64FX hardware, but improves the user experience

ABSTRACT

Arm HPC has succeeded in scaling up in the supercomputing space with the deployment of systems like RIKEN's Fugaku supercomputer and Sandia's Astra cluster. At the same time, the onboarding of new users to the Arm HPC ecosystem has never been more complex due to an overabundance of compilers, libraries, and build options for tools and applications.

This work investigates one particular method to ease the integration of new users into the space of Arm HPC through the use of Open OnDemand to provide a consistent and easy-to-use front-end for Georgia Tech's A64FX cluster, Octavius. We detail the motivations for this deployment as well as the potential pitfalls in integrating with an Arm A64FX environment. User-motived applications that incorporate the interactive usage of virtual desktops and Jupyter notebooks are discussed as motivating user workflows, and we provide some context on how future deployments might look with combined Arm and Open OnDemand integration.

CCS CONCEPTS

• **Hardware** → **Emerging tools and methodologies**; • **Software and its engineering** → *Open source model*.

KEYWORDS

cluster computing, user experience, Arm high-performance computing, Open OnDemand

ACM Reference Format:
Aaron Jezghani, Kevin Manalo, William Powell, Jeffrey Valdez, and Jeffrey Young. 2022. Onboarding Users to A64FX via Open OnDemand. In *International Conference on High Performance Computing in Asia-Pacific Region Workshops (HPCAsia 2022 Workshop), January 11–14, 2022, Virtual Event, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3503470.3503479

HPCAsia 2022 Workshop, January 11–14, 2022, Virtual Event, Japan

ACM ISBN 978-1-4503-9564-9/22/01…$15.00
https://doi.org/10.1145/3503470.3503479

1 INTRODUCTION

The past two years have seen the culmination of over ten years of effort to bring Arm-based platforms to the forefront of high-performance and high-throughput computing. From initial investigations into Arm HPC with the MontBlanc project [Rajovic et al. 2016] to the intense codesign efforts on the A64FX processor for the development of the Fugaku supercomputer [Sato 2020], there has been a concerted effort to develop both a strong hardware architecture as well as the tools and libraries to support applications at scale. In the high-throughput computing space, the introduction of custom processors like Amazon's Graviton [Jiang et al. 2020] and associated cloud-oriented tools have also reinforced Arm's place for both cluster installations and datacenter applications.

While Arm HPC has arguably succeeded in terms of supporting large-scale scientific and enterprise applications at scale, there are still some stumbling blocks that exist for growing the Arm HPC community and ecosystem. Notably, we find that new users are often confused about which compiler configurations and libraries work best for a particular application, and they struggle with finding the best setup for graphical applications or machine learning.

This work details and summarizes our efforts to deploy Open OnDemand [Chalker et al. 2021; Hudak et al. 2016] on our local A64FX cluster, Octavius, in Georgia Tech's novel architecture testbed, the Rogues Gallery. Specifically, we present a new workflow for admins and users that enables 1) seamless integration of Jupyter notebooks with HPC-style workflows, 2) the usage of standard desktop applications like Allinea's Map from a web browser interface for A64FX investigations, and 3) relatively simple integration with other state-of-the-art techniques like Singularity containers to support AI workloads via PyTorch and Tensorflow.

2 THE OCTAVIUS TESTBED

Our sixteeen node A64FX cluster at Georgia Tech is named "Octavius" due to it having a surplus of "Arms" like the roguish Doc Ock from some well-known comic books. Octavius is part of the NSF-funded novel architecture testbed, the Rogues Gallery [Young et al. 2019] and is the protoype piece of a larger NSF cluster at Georgia Tech called Hive. While this cluster cannot be expected to provide scaling on the order of other A64FX systems like Fugaku, Bristol's Isambard 2 cluster, or University of Stonybrook's Ookami testbed [Bari et al. 2021], it does provide local users with a very important prototype cluster to test vectorizable scientific applications.

The Rogues Gallery is a center-based testbed that is focused on near-term "post-Moore" computing including next-generation HPC, neuromorphic, near-memory, and reversible computing amongst other topics. The PIs of this Center for Research into Novel Computing Hierarchies (CRNCH) work closely with Georgia Tech's larger high-performance computing organization, the Partnership for Advanced Computing Environments (PACE), which supports approximately 2000 machines and more than 2200 active users. The Rogues Gallery helps to prototype small, next-generation systems while PACE enables the large-scale deployment of cutting-edge HPC and HTC environments including the NSF funded Hive project [Georgia Tech 2021].

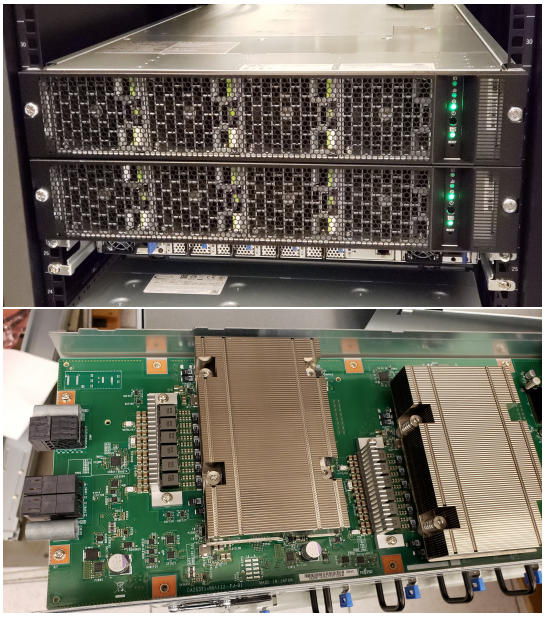

Figure 2: Octavius Testbed and Apollo80 Internals

The users of the Hive project were interested in having processors that are AVX-512 capable for code including genomics applications, astrophysics, and computational chemistry. Due to this interest, the addition of a small prototype cluster with the Scalable Vector Extension (SVE) and its 512 bit vector length is a natural extension for users familiar with the latest AVX-512 implementation.

One key issue for users interested to switch from a traditional Intel platform to A64FX is the recompilation of their code and linking of libraries. While Arm Performance Libraries [Arm 2021a] replicate much of the functionality of libraries like Intel's MKL, there is often a wide variation in performance and ease of use for compiling code with one of the four available compilers for A64FX SVE-enabled code: LLVM (and armclang), Cray Programming Environment (CPE), Fujitsu, and GCC.

Other migration sticking points for these users include general issues with developing a new workflow and finding the correct, optimized versions of certain packages to support their application. For example, Tensorflow and PyTorch can both currently be compiled for A64FX using related script and container recipes developed by Arm for Neoverse N1 [Arm 2021b], but the process may be complicated both for end users and system administrators looking to support these packages.

Our work with Open OnDemand looks to simplify that migration process for scientific users who would like to test SVE-enabled systems as an alternative to standard AVX-based platforms.

2.1 Architecture and Software

The Octavius testbed includes sixteen HPE Apollo 80 nodes, each with a 48-core Fujitsu A64FX processor, 32 GB of High Bandwidth memory (on-package), a 400 GB Micron 7300 M.2 SSD, and a Mellanox ConnectX-5 100 Gbps InfiniBand network card. All nodes are connected via a Mellanox EDR switch.

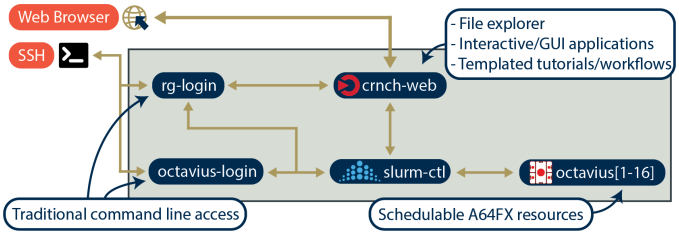

Figure 3: Octavius testbed workflow with Open OnDemand.

Our testbed setup is very similar to the larger Ookami cluster [Bari et al. 2021] deployed by Stonybrook University as an NSF tier-2 resource. Ookami is also based on the Apollo 80 platform, uses InfiniBand networking, and supports similar types of compilers (with the addition of Fujitsu's compiler, which GT does not support). We expect that insights for our smaller prototype are also applicable to larger clusters like Ookami.

2.2 Resource Management

Octavius is managed via Slurm scheduling with Slurm 21.08 as the default installation for A64FX nodes as well as our login node, which is hosted on an x86 VM. Since we do not have an aarch64 login node, users can select from a debug/compile queue to test and compile their code for A64FX and can then request a longer job to run their code at scale on the system. Shared home directories and NVMe-backed scratch storage are utilized throughout the Rogues Gallery testbed.

2.3 Open OnDemand

Open OnDemand [Hudak et al. 2016] is a web-based portal for HPC environments developed by Ohio Supercomputing Center (OSC) that was initially introduced in 2013 and has recently been improved through NSF funding and community engagement with other HPC sites. Open OnDemand integrates an Apache-based front-end with a NGINX architecture and support for the Ruby-based AweSim AppKit, which was also developed by OSC (and which differs from the similarly named Apple product).

Importantly, Open OnDemand allows for standardized user authentication and interaction with job schedulers, graphical applications, terminal sessions, and file explorers without requiring additional user plugins. This decoupling of the complicated back-end services from the user-facing interface provides an easy entry point for new users, as shown in Figure 3.

Most importantly for our goal of improving user access to A64FX, the Open OnDemand project allows for the deployment of specific notebook-based applications via templates, which can be shared via standard Git repositories. The applications deployed in Section 3 build on existing Open OnDemand repositories, and we have also created custom templates to support Arm-specific Jupyter notebooks that demonstrate SVE compiler comparisons, PyTorch, and Tensorflow jobs. Our configurations for deployment of Octavius Open OnDemand templates and all of our configuration files can be found as sub-modules at our public Github repository [Jezghani et al. 2021].

2.4 Open OnDemand Overheads

One potential concern with the usage of Open OnDemand is how this service will scale with a larger number of users and how a more interactive workflow affects the usability of the A64FX nodes themselves. The latter concern is reflective of the nature of interactive computing, whether through a browser-based portal like Open OnDemand or more traditional means such as SSH tunneling. However, this issue can readily be addressed via scheduler integration of Linux Control Groups (cgroups), which confine these applications to the requested resources on the A64FX nodes, thus minimizing impact on other users' work.

OSC reports that a virtual machine with 16 cores and 64 GB is more than enough to support 600 unique users each month and up to 60 concurrent Per User NGINX sessions [Center 2020], which may each host multiple notebook or application instances. Other recent work by this group demonstrates how their Open OnDemand instance has scaled to 200 remote students during the past year [Settlage et al. 2021] and how a more easily accessible front-end interface to HPC resources can potentially lower the barrier to entry. A 2018 study showed that adding Open OnDemand to traditional HPC resources lowered the average time from account creation to first login by 76 hours and decreased the average time from first login to first job submission by another 20 hours [Hudak et al. 2018]. With a complex set of resources like those in the Rogues Gallery, we anticipate that the usage of Open OnDemand, application-specific notebooks and templates, and seamless integration across A64FX and other resources would dramatically improve users' first interactions with the testbed.

From previous studies, we can assume that a similar VM configuration should be more than enough to host the typical CRNCH Rogues Gallery workload (currently up to 50 concurrent users at any one time) while an organization like PACE might require a slightly larger VM setup to host several hundred concurrent users.

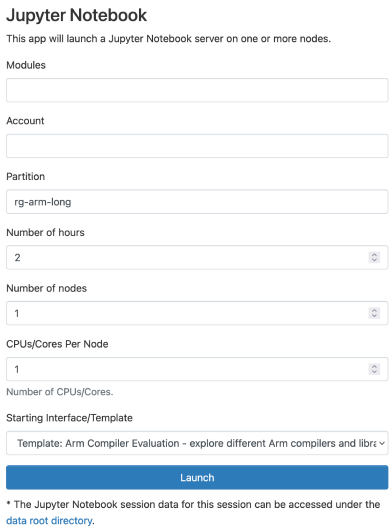

Figure 4: Open OnDemand webforms allow users to specify typical resource parameters and leverage templates for job submissions.

3 APPLICATIONS

Open OnDemand provides two main types of user interfaces to the cluster, Passenger applications and interactive applications. Passenger applications, which include the Files, Active Jobs, Job Composer, and Cluster Shell Access applications by default, typically run on either the web server or login nodes. Additional templates for clusters with XDMoD or Grafana dashboards are provided, but administrators can deploy custom applications using eRuby and javascript based on the available templates.

Interactive applications utilize the job scheduler to launch a persistent job on compute nodes and establish any necessary connections for browser-based access to the application. Job submission is streamlined via a customizable form, which can be as simple as free-form entry fields for each parameter, or more guided via dropdown menus and ranged inputs to align with compute hardware and policy. A basic job submission form is shown in Figure 4, where the inputs are coerced to the appropriate type, but users can enter any values.

Each interactive application utilizes its own form, so each application can have application-specific form fields for parameters including cores, initialized environment modules, paths, and environment variables. Once the application begins running on a compute node, the user can choose to launch their session, open a new terminal on the primary compute node, or review the job session files. Additionally, as can be seen in Figure 5, applications may provide additional runtime options, such as the VNC sliders to adjust video compression and quality to improve end-user experience.

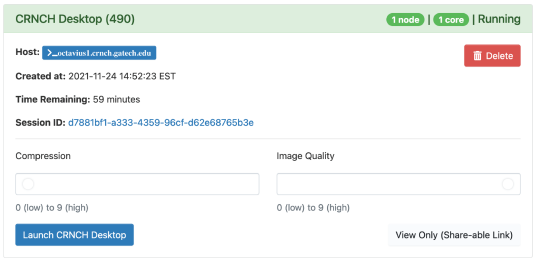

Figure 5: Users can connect to running applications from the interactive sessions screen

3.1 Virtual Desktop

The virtual desktop show in Figure 6 is one of the most straightforward and powerful applications for deployment on OOD. The application launches a virtual desktop via a batch job submission and then allows the user to connect in their browser via noVNC. In mirroring the templates available from OSC, we utilize an xfce desktop environment and TurboVNC for the virtual display server. While the xfce and libjpeg-turbo libraries are readily available from the EPEL repositories, each compute node requires the compilation of TurboVNC from source to produce an aarch64-compatible binary. The advantage of Open OnDemand here is that it allows us to use the same interface to launch interactive desktop jobs across both x86 and aarch64 compute nodes.

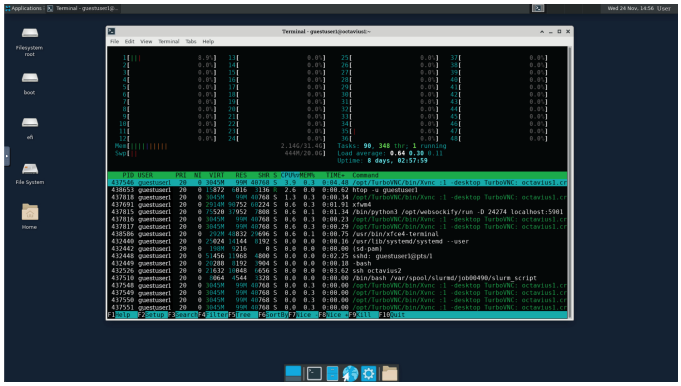

Figure 6: RG Open OnDemand virtual XFCE desktop. The OSC bc_desktop template was used with dependencies supported by aarch64 EPEL packages, aside from TurboVNC, which was compiled from source with the appropriate architecture flags.

In addition to the traditional "launch" button, the virtual desktop launcher includes compression and image quality sliders to provide better responsiveness for users with bandwidth limitations. Additionally, users can follow the "Session ID" link to open the Files application while a virtual desktop job is running, or click the job hostname to open an interactive shell on the compute node in the session directory.

3.2 Arm Map/Forge

Figure 7 demonstrates an example of running the HPCG benchmark on A64FX with Arm's Map tool as part of the Forge suite of tools. Normally, users would need to initiate a VNC session with port forwarding to connect to a profiling tool and run their application, but with Open OnDemand we can combine the batch scheduling for an interactive job, launching the Map application, and the loading of the appropriate modules and Arm compilers into one step for users. Applications that can operate as a web service (e.g., Jupyter notebooks, Intel VTune) can be executed directly in the browser in a lower latency fashion than via the virtual desktop. Even so, this setup with VNC still provides a easily accessible way to launch and use profiling tools that users might normally avoid with command-line or multi-step processes.

3.3 Jupyter Notebook

To reach our goal of improving usability with traditional HPC environments like our A64FX cluster, we look to use Open OnDemand to merge traditional HPC techniques (command-line compilation and analysis). As mentioned previously, Open OnDemand allows for the specification of user templates that can trigger the launch of different Jupyter notebooks. Currently, we support three different notebook-based templates: 1) an Arm compiler evaluation notebook based of Arm SVE tutorial material that shows how to integrate command-line compilation tasks and basic analysis 2) a Tensorflow notebook that uses Arm-developed Singularity containers to run a CIFAR example and 3) a PyTorch notebook using Arm-provided containers with CIFAR.

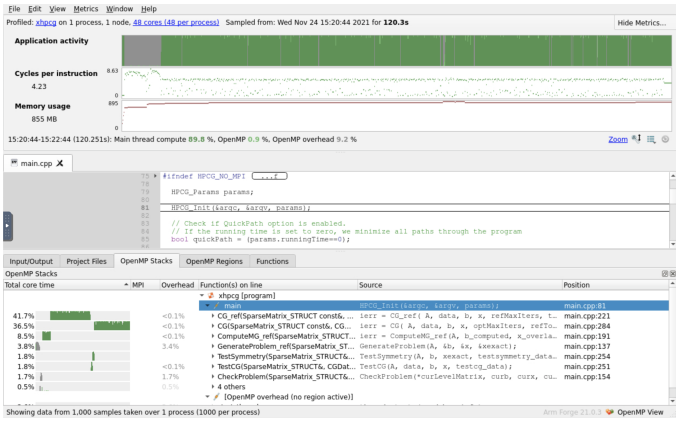

Figure 7: Browser-based access to an Octavius compute node running Map from the Arm Forge suite.

For the evaluation notebook, shown in Figure 8, we want to focus on the original workflow for running the Arm compiler examples in a shell by primarily relying on bash cell magic, and leveraging python exclusively for visualization. In order to simultaneously display the output for the user to review and make it accessible to python for manipulation, the bash output for each cell is teed to STDERR and subsequently processed into a new-line concatenated string of execution times. In each case, STDERR is then captured to a python variable via bash cell magic, and then transformed into a pandas dataframe for further exploration. Figure 8 shows part of the compiler evaluation notebook and how it combines standard command-line operations with Python-based data analysis.

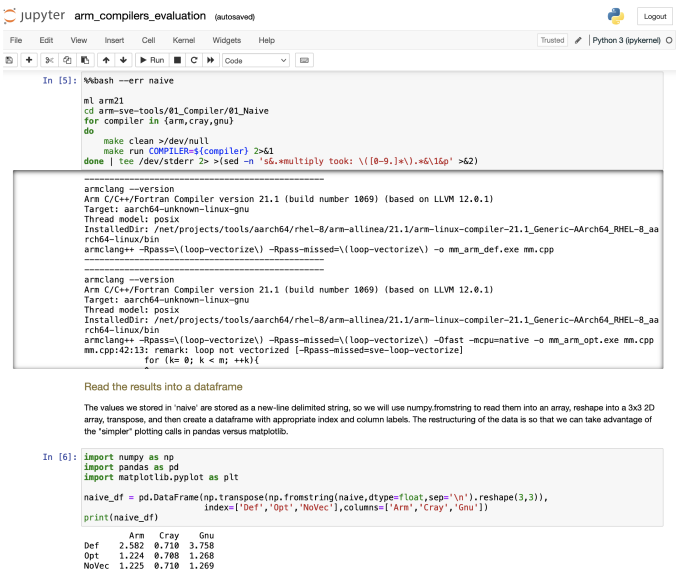

Figure 8: Arm compilers tutorial running in an Open OnDemand template notebook.

These notebooks for our A64FX templates and other Rogues Gallery resources can be downloaded by users from the public repository for the testbed [CRNCH 2021].

3.4 Tensorflow and PyTorch

Figure 9 shows a small snippet of PyTorch running a CIFAR training example on an A64FX node. From a backend standpoint, this notebook relies on Arm-provided Singularity containers and can take advantage of precompiled toolchains that use either OpenBLAS, Eigen, or OneDNN (Arm Compute Library) to accelerate ML training on Arm platforms. Similarly, Arm-provided TensorFlow containers can be used to quickly deploy AI training applications on A64FX, although we noted that the tuning of these containers for Neoverse N1 meant that some potential A64FX-optimizations were likely not included.

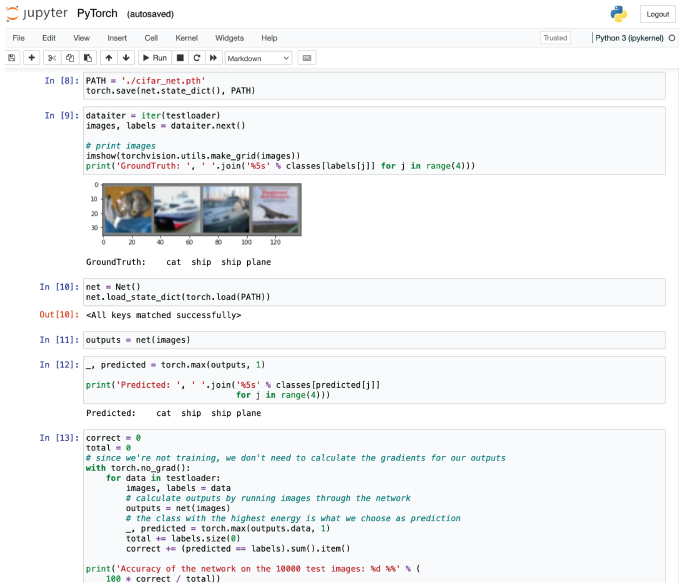

Figure 9: Snippet of a Jupyter notebook running a PyTorch Singularity container, launched through Open OnDemand.

4 CONCLUSIONS AND FUTURE WORK

The previous examples demonstrate how Open OnDemand allows us to seamlessly deploy workflows on an A64FX cluster as part of a larger testbed environment and how we can further engage users by abstracting the complexity of HPC and HTC workflows. In the current version of our testbed, users could potentially run part of their application on a Jupyter notebook, compile their code, launch a multi-node job on a set of A64FX nodes, and complete their data analysis all within the same environment.

Furthermore, the definition of Open OnDemand templates and applications using their AppKit-based approach allows us to define and extend sample workflows that users can pick up and utilize as new users to the system. This type of templating both improves the user experience and allows for more meaningful training opportunities in tutorials and related seminars that use the hardware.

Our future work will continue to extend the usage of Open OnDemand across the other types of novel architectures in the Rogues Gallery including GPUs, neuromorphic processors, and smart NICs. One might expect that a future scientific and AI workflow would incorporate the usage of a Jupyter-based workflow that calls out to A64FX for vectorizable kernels while targeting GPUs or spiking neural network accelerators for other GEMM-based computations.

Finally, we want to emphasize that much work still remains to improve the usability of these types of systems in the future. Recent Arm HPC workshops have demonstrated that there is wide variability in performance across compilers [Bari et al. 2021; Domke 2021], and build recipes for applications consistently pose problems for users and administrators alike. We hope that by coupling open-source build recipes (using scripting and/or Spack) and Open OnDemand templates that we can help to improve this process and to work towards a common set of best practices for current and future Arm-based clusters.

ACKNOWLEDGMENTS

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 testbed 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.

REFERENCES

Arm. 2021a. Arm Performance Libraries. https://www.arm.com/products/development-tools/server-and-hpc/allinea-studio/performance-libraries

Arm. 2021b. *Arm Tensorflow and PyTorch Docker Scripts.* Arm. https://github.com/ARM-software/Tool-Solutions/tree/tensorflow-pytorch-aarch64--r21.10/docker/pytorch-aarch64

Md Abdullah Shahneous Bari, Barbara Chapman, Anthony Curtis, Robert J. Harrison, Eva Siegmann, Nikolay A. Simakov, and Matthew D. Jones. 2021. A64FX performance: experience on Ookami. In *2021 IEEE International Conference on Cluster Computing (CLUSTER).* 711–718. https://doi.org/10.1109/Cluster48925.2021.00106

Ohio Supercomputing Center. 2020. *Open OnDemand Requirements (webpage).* https://osc.github.io/ood-documentation/master/requirements.html

Alan Chalker, Eric Franz, Morgan Rodgers, Trey Dockendorf, Doug Johnson, Doris Sajdak, Joseph P. White, Benjamin D. Plessinger, Mohammad Zia, Steven M. Gallo, Robert E. Settlage, and David E. Hudak. 2021. Open OnDemand: State of the platform, project, and the future. *Concurrency and Computation: Practice and Experience* 33, 19 (2021), e6114. https://doi.org/10.1002/cpe.6114 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6114

CRNCH. 2021. *CRNCH Rogues Gallery Notebook Repository.* https://github.com/gt-crnch-rg/sample-notebooks

Jens Domke. 2021. A64FX–Your Compiler You Must Decide!. In *2021 IEEE International Conference on Cluster Computing (CLUSTER).* IEEE, 736–740.

Georgia Tech. 2021. Georgia Tech Hive project. https://pace.gatech.edu/new-hpc-cluster-hive

David Hudak, Doug Johnson, Eric Franz, Alan Chalker, Troy Baer, Trey Dockendorf, and Katharine Cahill. 2018. Open OnDemand: Access Clusters, Gateways and Interactive Apps. *Lecture notes in networks and systems* (9 2018). https://doi.org/10.6084/m9.figshare.7069691.v1

David E. Hudak, Douglas Johnson, Jeremy Nicklas, Eric Franz, Brian McMichael, and Basil Gohar. 2016. Open OnDemand: Transforming Computational Science Through Omnidisciplinary Software Cyberinfrastructure. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale* (Miami, USA) *(XSEDE16).* Association for Computing Machinery, New York, NY, USA, Article 43, 7 pages. https://doi.org/10.1145/2949550.2949644

Aaron Jezghani, Kevin Manalo, Jeffrey Valdez, William Powell, and Jeffrey Young. 2021. *CRNCH Rogues Gallery Open OnDemand Repository.* https://github.com/gt-crnch-rg/crnch-ood

Qingye Jiang, Young Choon Lee, and Albert Y. Zomaya. 2020. The Power of ARM64 in Public Clouds. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID).* 459–468. https://doi.org/10.1109/CCGrid49817.2020.00-47

Nikola Rajovic, Alejandro Rico, Filippo Mantovani, Daniel Ruiz, Josep Oriol Vilarrubi, Constantino Gomez, Luna Backes, Diego Nieto, Harald Servat, Xavier Martorell, Jesus Labarta, Eduard Ayguade, Chris Adeniyi-Jones, Said Derradji, Herve Gloaguen, Piero Lanucara, Nico Sanna, Jean-Francois Mehaut, Kevin Pouget, Brice Videau, Eric Boyer, Momme Allalen, Axel Auweter, David Brayford, Daniele Tafani, Volker Weinberg, Dirk Brömmel, René Halver, Jan H. Meinke, Ramon Beivide, Mariano Benito, Enrique Vallejo, Mateo Valero, and Alex Ramirez. 2016. The Mont-Blanc Prototype: An Alternative Approach for HPC Systems. In *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis.* 444–455. https://doi.org/10.1109/SC.2016.37

Mitsuhisa Sato. 2020. The Supercomputer "Fugaku" and Arm-SVE enabled A64FX processor for energy-efficiency and sustained application performance. In *2020 19th International Symposium on Parallel and Distributed Computing (ISPDC).* 1–5. https://doi.org/10.1109/ISPDC51135.2020.00009

Robert E. Settlage, Alan Chalker, Jeff Ohrstrom, Eric Franz, Doug Johnson, and David Hudak. 2021. Open OnDemand as a Platform for Virtual Learning in Higher Education. *Lecture notes in networks and systems* 216 (2021).

Jeffrey S. Young, Jason Riedy, Thomas M. Conte, Vivek Sarkar, Prasanth Chatarasi, and Sriseshan Srikanth. 2019. Experimental Insights from the Rogues Gallery. In *2019 IEEE International Conference on Rebooting Computing (ICRC).* 1–8. https://doi.org/10.1109/ICRC.2019.8914707