Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing

Richard Lawrence[1]
rarensu@tamu.edu

Tri M. Pham[1]
phamminhtris@tamu.edu

Phi T. Au[1]
thau@tamu.edu

Xin Yang[1]
karen890@tamu.edu

Kyle Hsu[1]
hsuk1001@tamu.edu

Stuti H. Trivedi[1]
stutitrivedi1373@tamu.edu

Lisa M. Perez[1]
perez@tamu.edu

Dhruva K. Chakravorty[1]
chakravorty@tamu.edu

ABSTRACT

Successful outreach to computational researchers for informing about the benefits of switching to a different computing environment depends on the educator's ability to showcase practical research and development workflows in the new computing environment. Interactive, graphical computing environments are crucial to engage learners in computing education and offer researchers easier ways to adopt new technologies. Interactive, graphical computing allows learners to see the results of their work in real time, which provides the needed feedback for learning and enables chunking of complex tasks. Moreover, there is a natural synergy between computing education and computing research; researchers who are exposed to new computing skills within the context of an interactive and engaging environment are more likely to retain the new skills and adopt the new computing environment in their research and development workflows. Support for interactive, graphical workflows with modern computing tools in containerized computing environments has to be incorporated on high performance computing systems. To begin to address this deficiency, here we discuss our approach to teach containerization technologies in the popular integrated development environment of the Jupyter Notebook. We report on our scheme for implementing containerized software environments for interactive, graphical computing within the Open OnDemand (OOD) framework for research computing workflows, providing an accessible on-ramp for researchers transitioning to containerized technologies. In addition, we introduce several quality-of-life improvements for researchers and educators that will encourage them to continue to use the platform.

[1] High Performance Research Computing, Texas A&M University, College Station, TX

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/9

CCS CONCEPTS

•**CS**→Computer Science; •**Cybertraining**→training on using cyberinfrastructure; •**HPC**→high performance computing • interactive computing • training • containers

Keywords

Open on demand, Cyberinfrastructure, Portal, Jupyter notebooks, Singularity, Containers, Training

1. INTRODUCTION

Cyber Infrastructure (CI) is challenging; a complex computing environment is a barrier to learning for beginning researchers. There is an urgent need to offer scaffolded techniques in computing environments to reduce the barriers toward adoption of computing approaches. Making the computing environment engaging and more accessible provides opportunities for teaching in that environment while simultaneously improving user adoption of new technologies.

While the development of scientific computing applications and analysis of scientific data continues to be done on the command line, a broad swath of researchers by and large prefer interactive, graphical interfaces where they can develop their applications and visualize their data at the same time. An extremely popular example is the Jupyter Notebook [1] integrated development environment for Python users. Many important scientific research tasks are done in Python due to its ease of use in rapid development. For example, the popular artificial intelligence and machine learning (AI/ML) frameworks Keras, TensorFlow, and Torch are all Python language compatible frameworks. Although the workload for machine learning would most likely be distributed across a cluster, development is done interactively. This enables researchers to quickly debug their code by visualizing its output. The application in its final form would be exported to a text file for use in a batch script. If the researchers wished to deploy their application in a specific computing environment, then it makes sense that they would want to be developing and debugging applications in that same environment.

Powerful AI/ML tools such as the above are being widely adopted by the research community, as are the standards for research and development, FAIR (Findability, Accessibility, Interoperability, and Reusability) and FEAT (Fairness, Ethics, Accountability and Transparency). However, adoption of CI technologies and data frameworks that are needed to effectively support those standards

are lagging behind the adoption curve. As a result, educators lack the tools to properly teach AI/ML skills. This has hindered the adoption of FEAT and FAIR community standards for research. We need to find ways to make it easier to teach researchers how to use these technologies.

Education for research computation is well-served by interactive, graphical environments, which follows from the same reasoning. Researchers learning to use Python or any of the scientific Python libraries are more likely to retain knowledge and continue using the platform if it is the platform that will be used for research and development workflows. The Jupyter Notebook is well-suited for both Python education and development. The ability to visualize code, inputs, and outputs at the same time makes the integrated development environment a must for Python education. Cluster administrators are well-motivated to support the Jupyter notebook and other interactive and graphical computing interfaces to lower the barrier of entry for future researchers.

The Open OnDemand (OOD) platform provides a framework for supporting interactive and graphical environments [4]. It provides a Web interface to the computing cluster, which is advantageous because Web interfaces are generally platform-independent, which is good news for users working remotely from their personal machines. OOD interfaces on its back end with the computing cluster's batch system. Users can submit and monitor their batch jobs using an interactive graphical interface. The template for the job management interface is customized for each cluster. OOD also provides a framework for creating additional Web servers in the form of batch jobs assigned to compute nodes in the cluster. OOD allows the user to conveniently connect through a Web-based interface to this personal Web server. This is useful for a wide variety of graphical applications that can be served to Web browsers, from the humble Jupyter Notebook to as much as an entire virtual desktop.

The OOD platform installed out-of-the-box can be enhanced with features for educators and researchers. Its primary function is to support specific tasks but not necessarily to provide a comprehensive suite of tools to support a more general workflow. This is disadvantageous for educators who wish to quickly and effectively on-ramp researchers to the cluster for two reasons. First, if the workflow task the educators wish to teach isn't supported by the OOD platform, then they may need to use unrelated platforms for their teaching, which confuses researchers and leaves them with no clear transition to the development phase. Second, if researchers need to continue using command-line interfaces for some aspects of their workflow, then the benefit of adopting the OOD platform is marginal, which limits their openness to future education delivered through the platform.

One of the most important technologies for compliance with these principles is the container, a technique for bundling a software environment for maximum portability. A container allows the bundled software to run on any machine with the same basic architecture. The most popular container technology is Docker [5], which provides a file format for bundling software environments and a public repository named DockerHub for sharing those files. Many important scientific research frameworks, such Bioconda, LAMMPS, and Tensorflow, have already been containerized in the Docker format, allowing researchers working at home quick access to scientific software. However, for security reasons, Docker is not suitable for use on high performance clusters where many users share a filesystem. A solution to this problem is Singularity [6], which is an alternative to Docker that can both read the Docker file format and safely operate in cluster environments. Containers are clearly an important technology for the future, yet support for containers and especially support for container education lags behind the curve.

Adding container support to existing interactive, graphical interfaces will help educators better demonstrate container usage, showcase the utility of the approach, elevate the level of discourse to more advanced topics like reproducibility in computational sciences, and ultimately make it easier for researchers to navigate the transition to containerized computing environments. Alongside this, cluster administrators can also support sharing locally developed containers with external researcher collaborators. In the long run, these strategies will lead to a greater rate of container adoption and adoption of the FAIR and FEAT principles for scientific research, especially in the fields making use of biologically-inclined and AI/ML tools.

2. METHODS

In order to facilitate computational skills training for researchers, an interactive mechanism was needed. Open OnDemand offers that, but it is limited in its general scope to Jupyter notebooks and apps that represent the desktop. These were used for some educational courses. In addition, quality of life improvements were developed and support for containers was implemented. These features were also used for some educational courses.

2.1 Experimental Design

Open OnDemand has been used as an educational platform in several Short courses and Primers [2] since 2019. Students were polled for feedback after each to determine the achievement rate for the educational goals of each course. Results from this program have been reported previously [3].

2.2 System Design

At Texas A&M, several quality-of-life improvements and add-ons were implemented in the Open OnDemand framework, seen in Figure 1. These generally fall under the scope of User Experience, which is to say, how easily the user can get what they need from the platform. This is especially important for researchers who aren't necessarily trained in cluster computing technologies but still want to take advantage of those resources. These features help to draw researchers to the platform, which makes them more receptive to education delivered through the platform.

The helpdesk ticket submission form is a tool that allows researchers who encounter problems to request help from the cluster administrators by populating a form conveniently located in the OOD web interface. This makes it easier for researchers to get help since they don't have to leave the website to find instructions for how to do so. It also helps researchers get quick responses from cluster administrators because the form fields encourage the user to provide the administrators with the information needed to understand the problem.

The job composer form is a tool that allows researchers to edit batch scripts and submit them to the cluster by populating a form conveniently located in the OOD web interface. Parameters are selected in fields, while a preview shows the script that results from those parameter choices. Templates are provided for common scientific programming applications such as R and MATLAB. This makes it easier for educators to showcase the basics of batch scripting without the burden of also teaching command line syntax. It also enables researchers to get started quickly with their research tasks.

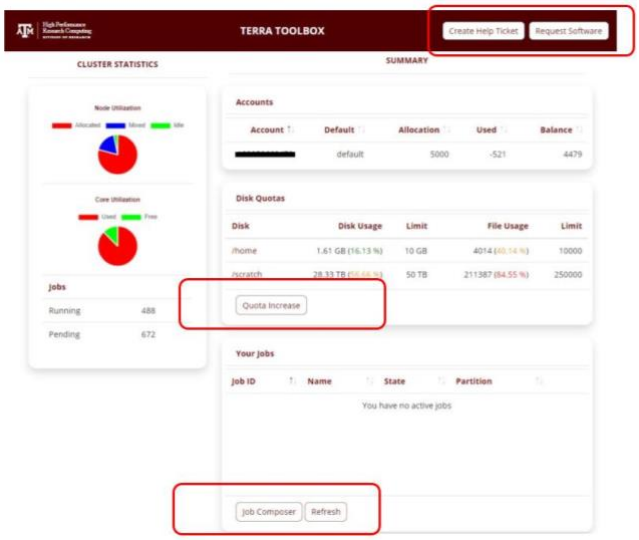

Figure 1. Screenshot of Dashboard page from Terra cluster maintained by Texas A&M High Performance Research Computing (HPRC). Tables shown: Disk Quota, Allocation, and Job monitoring. Highlighted: links to Create Help Ticket, Request Software, Quota Increase, and Job Composer forms.

The quota request form is a tool that allows researchers to ask for changes to their disk space allocations by populating a form conveniently located in the OOD web interface. This is a simple task, but researchers need to do it so rarely that when it comes time for it, they are often confused. Streamlining this process helps save time for researchers and cluster administrators.

The view-only link feature for the VNC interactive app is a tool for screen sharing implemented in the OOD framework. This is used by educators to provide instant feedback while teaching interactive computing skills. Each interactive job in the OOD framework creates a web server on a compute node and connects that web server to the student's browser. The student uses that web interface to perform activities on the compute node. In the case of VNC, the web server can also provide a second copy of the web interface that is read only. A link to this read-only web server can be distributed to allow teachers and collaborators to watch live as the job owner interacts with the compute node. In the case of VNC, this is useful for education of scientific computing skills for graphical software applications that display in a window, which is the majority.

A scheme for supporting containerized environments within OOD was developed. Support for containers can be added to any OOD interactive app simply by inserting the correct container runtime syntax in the template job script. In the case of the Terra cluster at Texas A&M University, the container runtime is Singularity [6]. For example, if the command to launch a Jupyter Notebook application that appears in the job template takes the form "jupyter notebook [arguments]," then the new syntax would be "singularity exec [image] jupyter notebook [args]." This substitution is relatively trivial, which is exactly why containers are such an important technology for clusters to support.

The major hurdle for implementing containerized app support in OOD is not the container syntax itself but the integration of that syntax into a larger structure that is flexible enough to support multiple computing environments. Although containers are fantastic, they are not for everyone. Cluster administrators would wish to continue maintaining their local computing environments to meet their users' needs. OOD does not provide a significant example or tutorial on how to go about supporting multiple computing environments. However, the underlying technologies are powerful enough to enable the necessary developments.

The OOD user interface for each app is assembled from a minimal list of specifications, which is convenient for simple apps, but the available options for customizing that user interface through the specifications are limited. The most significant limitation is that every field and widget with which the user may or may not need to interact must be displayed. In the case of multiple computing environments, where the information that is needed to select an environment varies greatly depending on which environment is to be used, this quickly grows out of control, hindering usability. The solution to this problem is to implement the OOD optional feature to support JavaScript in the user interface. This allows for the addition of scripts for toggling the visibility and values of the fields based on the content of the fields as shown in Figure 2.

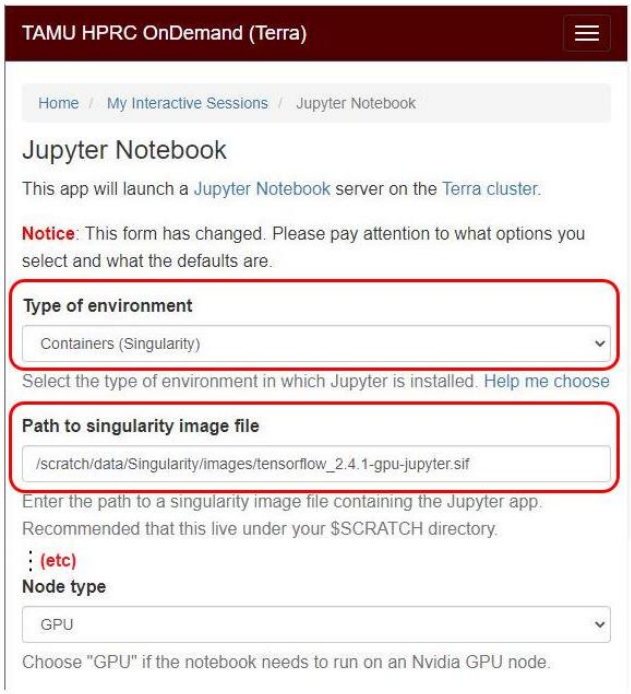

Figure 2. Screenshot of the Jupyter Interactive App form from the Terra cluster at HPRC. Highlighted: Drop-down menu for selecting environment type, Field for selecting container file.

3. RESULTS

Open OnDemand has been used by Texas A&M High Performance Research Computing (HPRC), and Laboratory for Molecular Simulation (LMS) for research computing since 2018. Results for using OOD in our introductory courses during 2020 have already been reported [3]. With over 1600 active users on the platform, several courses now use the Texas A&M OOD platform as the preferred teaching method. This includes a variety of advanced scientific computing skills in addition to the basic topic previously studied, as shown in Table 1. An important feature that accompanies this growth is the view-only link for VNC sessions, which enables a huge number of scientific applications that are compatible with VNC. However, the real winners are the many scientific applications based on the Python language, all of which

can now be demonstrated using the same hardware and software environment the researchers will ultimately use. The educational benefits of an interactive computing Web interface have made themselves evident through rapid adoption by instructors at Texas A&M HPRC as shown in Table 1. The view-only link VNC feature of OOD was also used during the instruction of CHEM 119: Fundamentals of Chemistry I, a formal laboratory course taught in Fall 2020 at Texas A&M. Ninety-five students practiced their computational skills as part of the course.

Table 1. List of informal HPRC short courses and Primers in Spring 2021. New OOD features that were used in the course and the number of participants attending. Courses continue to be offered on these topics.

Course Title	OOD Features	Partici-pants
Linux Primer	Shell access	110
Schedulers Primer	Job composer	292
Jupyter Notebook Primer	Jupyter Notebook	126
Cluster Usage Primer	Quota request form	203
Introduction to Python	Jupyter Notebook	235
Scientific Python	Jupyter Notebook	28
Introduction to Containers	Container support in Jupyter Notebook	4
Molecular Dynamics — NAMD	View-only link VNC	6
Introduction to Quantum Chemistry Simulations with ORCA	View-only link VNC	17
Drug Design — COVID19 for the cure	View-only link VNC	12
Intro to Xarray and Dask	JupyterLab Notebook	13
Intro to Deep Learning with Pytorch	JupyterLab Notebook	11
Intro to Deep Learning with Tensorflow	JupyterLab Notebook	19

3.1 Case Study Teaching Containerization on Clusters

The newest educational offering, Introduction to Containers [7], featuring OOD support for Containerized Jupyter Notebooks, demonstrates the importance of this ongoing effort. Introduction to Containers was a short course of two-and-a-half hours offered on April 30, 2021, October 15, 2021, and March 11, 2022. It was developed for the purpose of educating researchers about the use of containers. Due to COVID-19 conditions, the course was developed to be delivered virtual-only via Zoom.

The intended learning objectives were to:

1. Provide the researchers the information they need to decide if it is worth investing their time into making the switch to using containers in their research computing environment.
2. Familiarize the researchers with basic container workflow tasks, including use of the Singularity runtime, utilization of Docker-format container repositories, and support for containerized interactive graphical applications.
3. Demonstrate advanced applications of containers across multiple scientific disciplines to appeal to a wide audience.

To accomplish this, the course began with an overview of container technologies, leading into interactive exercises with which the students could follow along, and finally showcasing a selection of advanced topics. Learners were offered the materials from the course (for future reference) via the course website and our wiki. Wiki entries were updated to reflect the contents of the course.

The course had a number of descriptive examples for students to observe. These were followed with hands-on examples that allowed the participating researchers to practice their newly acquired knowledge. The Open OnDemand platform helped to make this course a success. Easy access to the cluster compute nodes via the OOD web interface removed the need for students to install SSH software on their local machines. Although command-line usage was used for some exercises, the equivalent batch script variations of those workflows were also demonstrated, which are entirely compatible with the OOD Job Composer feature. In principle, the researcher need not use the command line at all for those workflows. Finally, container support for interactive applications such as the Jupyter Notebook gave researchers the comfort of knowing that they could continue using their favorite graphical workflows should they choose to make the switch to containers.

The course was evaluated by polling the students for feedback at the end of the course. The poll was delivered through Zoom during the course and also by Google Form after the end of the course. Although there were few responses (due to the low attendance), the responses were overwhelmingly positive, indicating a high rate of success for the first two learning objectives. Students indicated that the course did give them the information they needed to decide whether to make the switch to containers and that they had gained familiarity with the basic workflows. The feedback indicated that the advanced topics were not of interest to most students, which leads us to the conclusion that the course can also be shortened by removing the advanced topics and offering it as a 1-hour primer. This could potentially increase interest in the course by decreasing the time commitment and by allowing the course to be offered at different times. We further surmise that offering the course during the last week of the semester had limited the number of participants in the course.

The most important improvement that is needed for the introduction to containers course is advertisement to counter the low attendance rate of the first course offering. Now that the course has been shown to be successful, researchers should be made aware that container education is an option for them. One way to deliver this advertisement is to subtly (or not-so-subtly) place visible evidence of container support throughout the OOD web interface as the support is expanded. This will help with recognition among the large audience of researchers who have been using the OOD platform exclusively since the success of the basic Primers and Short Courses of 2020.

A major benefit of the containers course is the cross-topic learning gain. Researchers learning how to use containers are by definition exposed to the principles of sharing and portability that led to the creation of the container technology. This helps to drive long-term adoption of FAIR and FEAT practices in research computing. Knowledge of containers in general is relevant for scientific research, especially in the fields making use of AI/ML tools.

4. CONCLUSIONS AND FUTURE DIRECTIONS

Since 2018, Texas A&M HPRC and LMS have successfully used the OOD platform to offer informal instruction at week-long summer schools, online Primers, and our short course program. Here, we reported on our efforts while highlighting a containerization course and Primer that benefitted from expanded capabilities added to the Texas A&M OOD framework. The use of OOD in the course, coupled with enhanced usability features, help increase learner engagement, improve adoption of the technology, and encourage continuous use of the technology in future research. Several of these features are developed and maintained by undergraduate and graduate students in the Texas A&M HPRC student fellows program. While singularity was the chosen containerization technology, the course offered a mechanism for researchers to use a Docker alternative. Recognizing the challenges faced in evaluating virtual courses, this course has been offered in a hybrid face-to-face and virtual setting with the goal of collecting more statistics on researcher usage and learning. We hope to further the learning gains by improving the rich feature set on our local Texas A&M OOD by including access to our knowledge base and ensuring access to a locally hosted container registry.

5. SUPPORTING INFORMATION

All training materials used in this study are available to the community via the Texas A&M HPRC website [2]. Videos and course recordings may be accessed via the Texas A&M HPRC channel on YouTube. The community is invited to join the SWEETER Slack workspace at https://hprc.tamu.edu/sweeter. Surveys and review exercises that will be developed as part of this longitudinal study may be requested from the author. Please send us feedback about your adoption experience via an email to help@hprc.tamu.edu.

6. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (NSF) award number 1925764, “CC* Cyberteam SWEETER,” and NSF award number 2019129, “MRI:FASTER”, NSF award number 1730695, “CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students”, NSF award number 2019136, “CC* BRICCs: Building Research Innovation at Community Colleges,” and NSF award number 1829799, “Cybertrainining: CMS3”.

7. REFERENCES

[1] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing and Jupyter development team. 2016. Jupyter Notebooks — a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Fernando Loizides and Birgit Scmidt (Eds.), 87–90. DOI: https://doi.org/10.3233/978-1-61499-649-1-87

[2] Texas A&M High Performance Research Computing Website. Retrieved from https://hprc.tamu.edu

[3] Dhruva K. Chakravorty, Lisa M. Perez, Honggao Liu, Braden Yosko, Keith Jackson, Dylan Rodriguez, Stuti H. Trivedi, Levi Jordan and Shaina Le. 2021. Exploring Remote Learning Methods for User Training in Research Computing. *Journal of Computational Science Education* 12, 2 (Feb. 2021), 11–17. DOI: https://doi.org/10.22369/issn.2153-4136/12/2/2

[4] Dave Hudak, Doug Johnson, Alan Chalker, Jeremy Nicklas, Eric Franz, Trey Dockendorf and Brian L. McMichael. 2018. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3, 25 (May 2018), 622. DOI: https://doi.org/10.21105/joss.00622

[5] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 2014, 239 (Mar. 2014), 2. Retrieved from https://dl.acm.org/doi/10.5555/2600239.2600241

[6] Texas A&M HPRC wiki page for Singularity. Retrieved from https://hprc.tamu.edu/wiki/SW:Singularity

[7] Texas A&M HPRC Containers Short Course Website. Retrieved from https://hprc.tamu.edu/training/intro_containers.html