ORIGINAL ARTICLE

Journal Section

Open OnDemand: State of the Platform, Project and the Future

Alan Chalker PhD[1] | Eric Franz[1] | Morgan Rodgers[1] | Trey Dockendorf[1] | Doug Johnson[1] | Doris Sajdak[2] | Joseph P. White PhD[2] | Benjamin D. Plessinger[2] | Mohammad Zia PhD[3] | Steven M. Gallo PhD[3] | Robert E. Settlage PhD MS[4] | David E. Hudak PhD[1]

[1]Ohio Supercomputer Center, Columbus, OH, 43212, USA

[2]Center for Computational Research, University at Buffalo, Buffalo, New York, 14203, USA

[3]Roswell Park Comprehensive Cancer Center, Buffalo, New York, 14203, USA

[4]Advanced Research Computing, Virginia Tech, Blacksburg, Virginia, 24060, USA

Correspondence
Alan Chalker PhD, Ohio Supercomputer Center, 1224 Kinnear Rd. Columbus, OH, 43212, USA
Email: alanc@osc.edu

Funding information
National Science Foundation of the United States Awards NSF SI2-SSE-1534949 and CSSI-Frameworks-1835725.

High performance computing (HPC) has led to remarkable advances in science and engineering and has become an indispensable tool for research. Unfortunately, HPC use and adoption by many researchers is often hindered by the complex way these resources are accessed. Indeed, while the web has become the dominant access mechanism for remote computing services in virtually every computing area, HPC is a notable exception. Open OnDemand is an open source project negating this trend by providing web based access to HPC resources (`https://openondemand.org`). This paper describes the challenges to adoption and other lessons learned over the three year project that may be relevant to other science gateway projects. We end with a description of future plans the project team has during the Open OnDemand 2.0 project including specific developments in machine learning and GPU monitoring.

KEYWORDS
Open OnDemand, Science Gateway, High Performance Computing, Interactive, Web Platform

Abbreviations: HPC, high performance computing; OOD, Open OnDemand; SCAG, Scientific Community Advocacy Group.

1 | PROJECT TIMELINE

At the XSEDE 2013 conference we introduced OSC OnDemand, OSC's web platform for providing OSC users web based access to HPC resources [1]. Through OnDemand users can create, edit and upload/download files, create, edit, submit and monitor jobs, create and share apps, run GUI applications and connect to a terminal, all via a modern and familiar web browser. Significantly, all of this is accomplished without user-side software or browser based plugins to install. Representatives from multiple HPC centers gave us the same feedback: "how do we run this at our center"? With this feedback, in 2015 we applied for and were granted an NSF award (SI2-SSE) to develop Open OnDemand [2], an open-source version of OSC OnDemand that other HPC centers would be able to install. The SI2 project ran from July 9, 2015 January 31, 2019.

The beta was made available in the fourth quarter of 2016 and announced via webinar on March 8, 2017. The beta supported file management, shell access and included a customizable dashboard with basic branding and display of a site's message of the day. Job management and monitoring was available only to sites using the Torque resource manager, however, none of the centers external to OSC that were initially interested ran Torque. Over the four months following the first webinar we added support for batch schedulers Slurm and LSF and worked with interested centers to install the platform on their systems. On August 3, 2017 we moved from the beta to 1.0. We had 5 more releases during the course of the project: 1.1 was released September 29, 2017; 1.2 on November 3, 2017; 1.3 on May 8, 2018; 1.4 on January 8, 2019; 1.5 on February 21, 2019. Each year of the project OnDemand had a presence at the XSEDE/PEARC, Gateways, and SC conferences which provided opportunities to both advertise the project and solicit feedback from the community.

A large part of the project was community engagement, which included a series of six interactive webinars starting in March 2017. From March 2017 to July 2018 the bulk of electronic community engagement was facilitated through a project mailing list. Because of the difficulty in tracking and searching through multiple threads on a mailing list, in July, 2018 we stood up a Discourse instance (http://discourse.osc.edu/) and started transitioning community discussion to the online forum. This, along with direct email, is now the primary community engagement method we use, as illustrated by the total of 108 distinct threads created on Discourse, consisting of 534 individual posts.

2 | PLATFORM CAPABILITIES

"The Open OnDemand 2.0 project, is an open-source software project that enables access to high-performance computing, cloud, and remote computing resources via the web" (NSF award 1835725) with the lofty goal of lowering barriers to HPC use for everyone. To accomplish this, OnDemand leverages HTML5 standards as a platform for development of a feature rich set of applications giving users a familiar web based front end to HPC systems. The use of HTML5 standards enable the use of a web proxy that provides a federated authentication ensuring the security of the system. OnDemand provides HPC centers a "zero-install" and single sign-on (SSO) solution for their users.

A typical workflow for an OnDemand user begins with the user opening a web browser and navigating to their HPC center OnDemand url. The user will then authenticate with their HPC credentials and be redirected to the OnDemand landing page. The landing page for OnDemand is the Dashboard App. This app is really the gateway to the various OnDemand features. It not only provides links to the default set of core OnDemand apps but also to any custom apps developed to extend OnDemand. These custom apps are being developed by the OnDemand team and by other HPC centers that have adopted OnDemand. By default, OnDemand includes a set of core apps for file management (as shown in Figure 1), job management (as shown in Figure 2) and monitoring and access to clusters via the command line. This set provides the bare minimum functionality of accessing and managing jobs and related files that allow a traditional client to utilize available HPC resources.

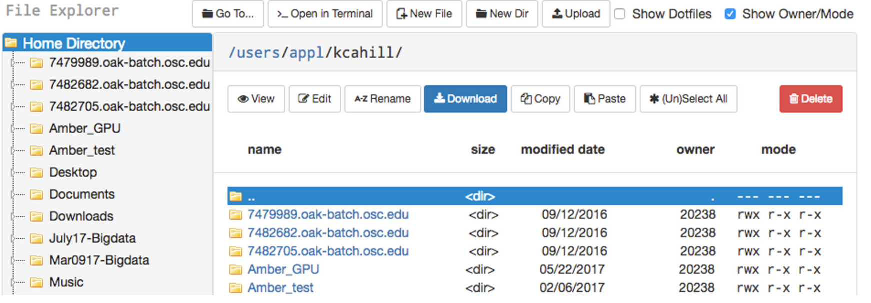

FIGURE 1 Open OnDemand's file explorer app allows users to work with files on the remote filesystem just like they can on their own desktop machines.

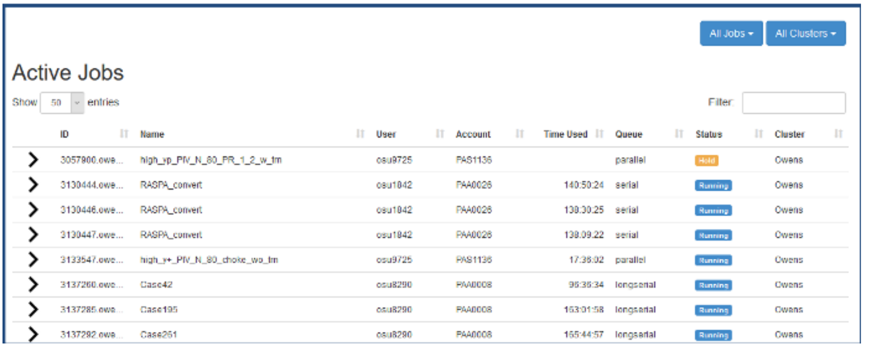

FIGURE 2 Open OnDemand's active jobs app allows users to view the status of jobs submitted to the HPC system's scheduler.

"A key feature of OnDemand over a traditional web service is the per-user web server model where all the web applications are run as the Linux user" as shown in Figure 3 and more fully described in [3]. Many traditional web services have a design requiring the core running applications to have elevated or root permissions on the server. Individual users' roles and permissions are managed within the application itself, which can be complicated and potentially result in significant security vulnerabilities. In contrast, OnDemand leverages the Linux kernel for security and accounting. This removes the onerous task of securing apps from the local developer, making OnDemand a great choice for deploying center specific user applications. Popular scripting languages including Ruby, Python, and Node.js are supported for app development. OnDemand also includes an "AppKit library which extends the capabilities of web applications by providing a unified interface to a variety of resource managers, including Torque, Slurm, LSF, and PBS Pro" [3].

OnDemand also supports Interactive Apps. An Interactive App consists of a web form for configuring job and application specific settings and a bash script for submission of a batch job that launches a web application server on a compute node (as seen in Figure 4). Various aspects of the job, such as a password-protected VNC server, are

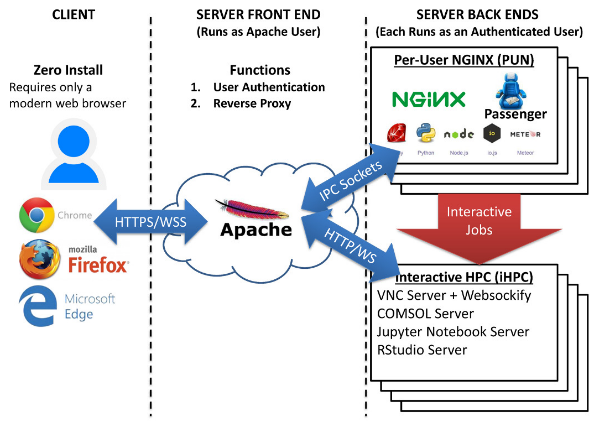

FIGURE 3 The Open OnDemand architecture includes key components such as Apache on the front end, and NGINX and Passenger on the back end.

pre-configured in such a way as to allow the user to securely connect directly to compute nodes from within their browser. These apps are often utilized for pre- or post-processing of data, which typically requires the user to directly interact with a graphical interface. Interactive Apps are the foundation of science (Jupyter Notebook or RStudio Server in Data Science; QGIS in map based science, etc). Sooner or later, a human needs to observe input or output to make a decision. To facilitate development and deployment, the Dashboard comes with a plug-in style wrapper for launching a standard HPC job, starting the application and serving the GUI through the web browser. OnDemand currently supports Jupyter Notebook (both with and without a standalone Spark cluster running within the batch job), RStudio Server, COMSOL Server, qGIS, ParaView, VisIT and a wide range of X11 GUI applications accessed through VNC (`https://osc.github.io/ood-documentation/master/install-ihpc-apps.html`.

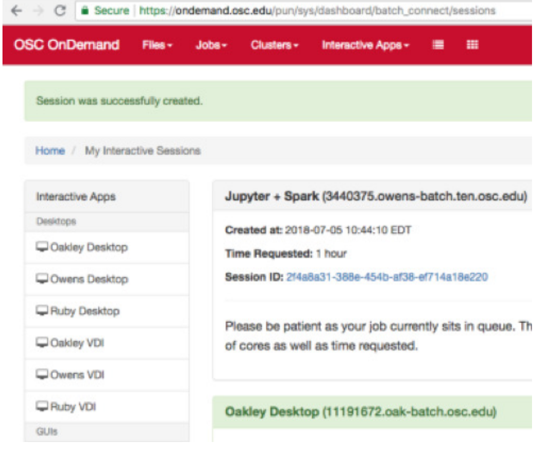

FIGURE 4 OnDemand's interactive app dashboard showing desktop and jupyter + spark sessions for the current user.

3 | ADOPTION

In 2016 there were 0 installs of the platform, in 2017 there were 10 installs, and in 2018 over 50 installs. Even though the software was publicly accessible in the fourth quarter of 2016, it wasn't until the March 8, 2017 webinar along with web accessible installation documentation that we started to see other sites install OnDemand. To help incentivize adoption we offered to directly support system administrators through the installation, and even do it ourselves if provided accounts on the remote systems. Indeed, the first centers to install OnDemand provided test accounts that also enabled us to add support for batch schedulers such as Slurm, LSF, and PBSPro.

To facilitate adoption, we provided Docker and Vagrant images containing a very basic deployment of OnDemand to help people preview a working (but non-production) version of OnDemand as part of their evaluation process. When requested, we also provided staff at other HPC centers accounts on OSC systems so they could be exposed to our production OnDemand deployment at OSC. The biggest impact to adoption was the introduction of an RPM based installation in the 1.3 release on May 8, 2018. By the October 17, 2018 webinar we had logged RPM installs from 50 distinct institutions in the US (many of which are shown in Figure 5).

FIGURE 5 Logos of some of the organizations with production deployments of the Open OnDemand platform as of early 2019.

As an indicator of OnDemand's impact, consider the Ohio Supercomputer Center's (OSC) OnDemand usage. In 2017, OSC OnDemand (1) had 1,112 users working on projects from 28 distinct fields of science, (2) helped new users accelerate their time to science by reducing the median time to submit an initial HPC batch job to one-tenth of the time compared to using traditional SSH (as shown in Figure 6), and (3) became OSC users' preferred mechanism for connecting to HPC as the number of OnDemand users exceeded the number of traditional SSH users.

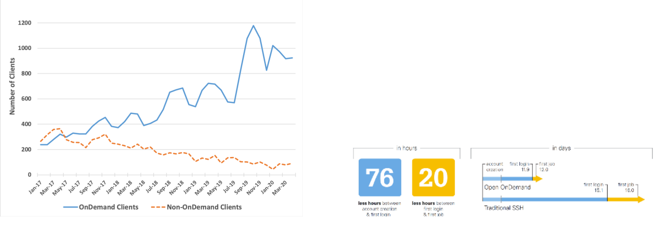

FIGURE 6 Left: Expansion of OSC OnDemand use. Right: Median times to science comparing using OnDemand versus traditional SSH access from 1712 OSC accounts created in 2017.

4 | LESSONS LEARNED

This project's success is due primarily to: 1) an unmet need to improve access (reduce barriers and gain efficiencies) to HPC resources across many disciplines and 2) our intentional engagement with the community to gather and respond to needs and feedback. If we did it over again, we would have prioritized ease of installation and configuration earlier on. These are some of the lessons we have learned throughout the first phases of this project.

4.1 | Engage the Community on Features

During the course of the 3 year project we gave six webinars. During each webinar we would ask for and receive feedback and reiterate the importance for the community to drive the project. As a result of actively supporting OnDemand installs, we developed relationships with system administrators at multiple institutions. We solicited feature ideas and prioritization directly from these administrators. Our website also prominently asks for feature requests and invites users to provide them on Discourse or GitHub. We have received such feedback through direct email, emails to the mailing list, feature requests to Discourse, and in the form of GitHub issues.

This feedback helped us focus efforts on what was important to the community. Based upon our experience at OSC, the project team originally thought building custom OnDemand web apps (form based job configuration and launcher, e.g. an OpenFOAM or BLAST+ launcher) was going to be the killer feature. While some centers ended up building custom web apps, this feature was not widely utilized. Instead, most centers saw the primary value of OnDemand as enabling users to launch interactive apps such as Jupyter, RStudio, and VNC desktop, and enabling users to do basic file management via the web browser. Based on direct feedback from users, we have added many features. Examples of community directed development includes: (1) expanding OnDemand's job adapter interface to support other batch schedulers such as LSF, PBSPro and SGE; (2) adding Xfce support for VNC desktops; (3) options for enabling two factor authentication; (4) ability to open PDF and HTML files in the Files app for easy viewing; (5) customizable error pages for the first login flow of a new HPC user whose home directory might not exist.

The combination of adding features that centers requested along with actively supporting center installs encouraged the growth of a community that is excited about OnDemand. However, our eagerness to quickly respond to community requests did result in acquiring some technical debt, which we are currently looking at addressing. Additionally, we have had to step back and review how we prioritize app development, feature building etc to support a growing and vibrant community in a way sustainable for our development team.

4.2 | Be Responsive to the Community

Sometimes it is not enough to just ask the community what they want. It is important to observe the research community's interests and trends. When the OSC OnDemand XSEDE 2013 paper was published, project Jupyter did not exist. But in the past few years, Jupyter has been rapidly adopted within the scientific community, to the point it is now the defacto standard. Likewise, RStudio has become extremely popular for anyone using the R language. As a direct result of the HPC community adoption of these applications (observed via client requests, usage at OSC and peer HPC centers, and citations in publications), in early 2016 we developed plans to support launching both Jupyter and RStudio, and these two apps became the most popular interactive apps in OnDemand.

4.3 | Lower the Barrier to Testing the Platform

It's important to make it easy to evaluate the software. Since installing OnDemand at a center is a non-trivial task, we provide both Vagrant and Docker images with OnDemand installed. One of the Vagrant images includes two nodes, one with OnDemand installed, and one with Slurm installed that also acts as a compute node. This enables testing the job management capabilities within OnDemand. For a more comprehensive evaluation we also provide interested users test accounts at OSC to try out our installation of Open OnDemand. While we do not have metrics to assess how effective these were in driving adoptions, we did have multiple centers evaluate Open OnDemand through test accounts at OSC and we also demonstrated OSC OnDemand directly. In both cases, these centers eventually installed OnDemand.

4.4 | Think about Installation and Configuration

If we started this project from scratch today we would plan on supporting RPM based installation from the start. Indeed, this is the goal driving the future design of OnDemand's architecture. Unfortunately we initially added support for installation at other centers using two strategies: (1) identify all OSC specific code and replace with configuration that can be added to the environment to custom Rails initializers; (2) write build from source installation steps, of which there were many. Some configuration was build time configuration: required prior to building the apps, which is a violation of the design principles that enable Continuous Delivery and RPM based installation. Code was spread over many git repos instead of being consolidated in one, increasing the number of installation steps.

Installation also required Apache configuration which would vary from center to center based on authentication requirements. We were surprised to learn that often the center's support staff, not systems staff, were installing OnDemand. These staff did not have expertise in Apache configuration and ran into many problems during the installation process.

If we had designed OnDemand with the intention of supporting RPM based installation from the start we likely would have fewer repos and would have designed runtime only configuration with proper separation of configuration and application code. Instead we had to make these changes in version 1.2 to support moving to RPM based installation in version 1.3.

4.5 | Streamline the User Interface

While OnDemand consists of several silo-ed apps, basic integration was attempted by implementing a URL "API interface" in the shell, files, and file editor apps. Specifically, the URL for the shell app can include the path to a directory to change to to when the ssh session starts; the files app can include a path to a directory to show; the file editor includes a path in the URL to the file to open in the editor. The result is that many views in OnDemand can provide contextual buttons to open relevant files or directories in the corresponding apps. This one click access saves time for users.

The OnDemand dashboard has an app discoverability feature where a web app or an interactive app plugin can be deployed to a directory by an administrator and a link will appear in the user's dashboard. There is no user side step necessary. This makes it easy for administrators to deploy new apps. This was utilized by centers to deploy or develop new interactive apps that OSC had not deployed at the time, including VMD and JupyterLab. In the process of managing app installations, sometimes an administrator will rename an app directory, with a suffix ".old" or ".1" in order to hide it. The dashboard skips over these directories when building the navigation menu. These two useful features are a reminder that the installation and configuration of a gateway is also part of the “user interface”.

4.6 | Centers want a Unified Interface

In discussions with several Center admins, we found they all want a unified administrative interface, aka a “single pane of glass” for their center. Requests for features of this interface include the ability to see current (Figure 7 for example) and historical resource utilization at a variety of levels (e.g. individual, group, system, etc.), as well as the ability to directly access the resources in a variety of ways (e.g. drag and drop file-system browsing, command line access, etc.) An example of a solution to this is that one center developed very basic ‘apps’ for their administrators that execute simple commands such as quota and disk utilization and display the nicely formatted output within the OnDemand browser window.

Center admins were quick to detail the specific tools they either envisioned or prototyped themselves. Many of these could be utilized by other centers if they were to be contributed back to the general code base. Others could be easily replicated and replaced by more sophisticated software, such as the NSF-funded Open XDMoD platform [4]. As this was beyond the scope of the initial project, it was included in the scope of the follow-on project discussed in Section 5.

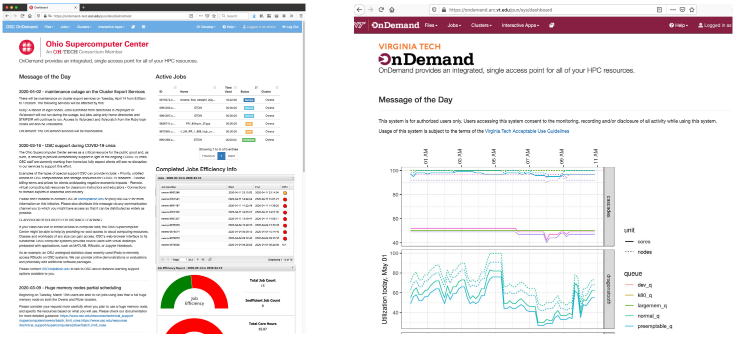

FIGURE 7 Ohio Supercomputer Center and Virginia Tech Open OnDemand landing pages showing local customizations.

4.7 | Think about Security

Since Center admins are generally by nature very security aware and averse to exposing their systems to un-tested system software, we regularly have been asked about the security of OnDemand. While the code is open-source and viewable by everyone, few people have the time to completely review it. Thus, we engaged with the NSF-funded Trusted CI Cybersecurity Center of Excellence to perform a security audit of the platform [5]. Trusted CI examined OnDemand in detail and provided the following summary: they "(1) discovered several implementation issues (bugs) that could affect the proper operation of Open OnDemand; (2) No major issues were found in a comprehensive evaluation of the architecture and critical resources in Open OnDemand, and analysis of potential code weaknesses in critical components; and (3) They found a potential vulnerability, dormant based on current configuration and launch settings". These findings provided us with some confidence in the security of the platform, as well as some issues to focus our efforts on resolving.

4.8 | Retain an Escape Hatch

Our ability to respond quickly to change requests from the community was made possible in part by the choice of using Phusion Passenger as our application server. Passenger supports Python, Node.js, and both Ruby on Rails as well as arbitrary Ruby web frameworks. Hence, OnDemand's architecture enabled us to serve multiple web apps in multiple interpreted languages, including Ruby for rapid web development. The combination of Ruby's metaprogramming capabilities and the Rails web application framework's initializer files, code that is executed at the start of a Rails application, provided us an escape hatch. Often when a center would ask for a feature or a bug fix, we would be able to provide that center a temporary fix in the form of Ruby code to add to an initializer in a subdirectory of the deployed app. In OnDemand 1.3, when we introduced RPM based installation, we added support for adding these custom initializers to subdirectories under /etc/ood/config. The temporary fix would utilize Ruby's metaprogramming capabilities to rewrite a method or add a new class to the application at runtime. A center could benefit from this temporary fix until the next OnDemand version containing the fix was released.

5 | FUTURE WORK

In 2018, OSC with partners Virginia Tech and SUNY Buffalo were awarded an NSF grant under the CSSI program managed by the Office of Advanced Cyberinfrastructure. This ca. $3.3M award (#1835725) provides funding through the end of 2023 and is formally titled "Frameworks: Software NSCI-Open OnDemand 2.0: Advancing Accessibility and Scalability for Computational Science through Leveraged Software Cyberinfrastructure" This new project combines two widely used HPC resources: Open OnDemand 1.x - an existing open-source, web-based project for accessing HPC services; and Open XDMoD - an open-source tool that facilitates the management of HPC resources.

The project has the following stated objectives:

- "Objective 1 (Visibility): Leverage XDMoD seamlessly from OnDemand, creating a unified platform for scientists to both work with HPC and optimize their HPC work."
- "Objective 2 (Accessibility): Improve the OnDemand interface for more scientists & fields of science."
- "Objective 3 (Scalability): Extend the scalability of OnDemand for more platforms and applications."
- "Objective 4 (Engagement): Conduct a program to engage departmental, campus and national HPC users along with the Science Gateway community to drive adoption and follow-on development."

Work is well underway now on each of these objectives, which includes: (1) Initial prototypes linking XDMoD graphs within OnDemand; (2) A survey of community opinions and feature requests resulting in scientific domain support expansion and platform extensions, and (3) Establishment of an advisory board and Community Engagement.

5.1 | XDMoD Integration

An important aspect of the OnDemand 2.0 project is integrating XDMoD cluster usage and job data and metrics. As novice users find themselves in an environment where multicore and multinode and perhaps multicluster computing are the expectation, tools to assist in job placement and monitor job efficiency are more important. XDMoD driven center state information gives users a guide to system utilization that may help in obtaining a quick or scaled job (Figure 8). Integration of the job metrics from XDMoD with the job submission and status apps gives users a quick assessment of how their jobs are performing (see OSC proposed landing page in Figure 7 and Figure 9). Active embedded links in the job status display allows users to drill down into linked XDMoD displays to more fully characterise a job. As new users gain experience, these tools can be further utilized to create benchmarks for later optimizations.

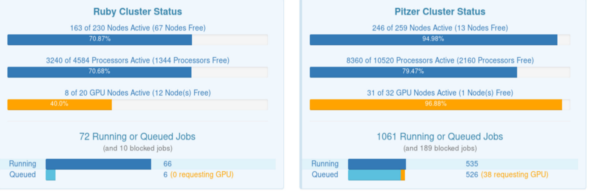

FIGURE 8 Prototype linking of Open XDMoD graphs for cluster utilization calling out GPU availability.

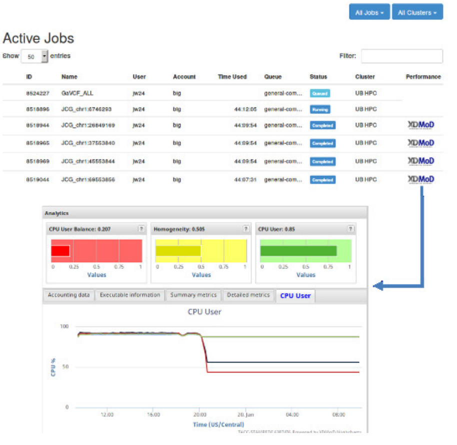

FIGURE 9 Prototype linking of Open XDMoD graphs from within the Open OnDemand active jobs app.

5.2 | Scientific Domain Coverage Expansion

A key aspect of the second phase in the Open OnDemand grant is making HPC accessible to more scientific disciplines. Engineering, fluid dynamics, computational chemistry, and computer science have all been indoctrinated into the HPC idioms of cores, nodes, schedulers, jobs etc necessary to compute on these large clusters. Through OnDemand, we are endeavoring to lower the barrier to entry in all fields with a computing need. In many cases, this means creating an app for the GUIs used in the field. For instance, in mapping applications - QGIS, genomics - Integrated Genome Browser, financial statistics - STATA and deep learning - Tensorboard (Figure 10).

5.3 | Platform Extensions

As more and more centers evaluate and adopt OOD, the need to facilitate the deployment of OOD on a variety of HPC resources has grown. IBM Power, Nvidia DGX and on/off-prem cloud platforms are all targets for alternative deployments. At Virginia Tech, Open OnDemand serves an IBM Power8 cluster (see Tensorboard app, Figure 10. This

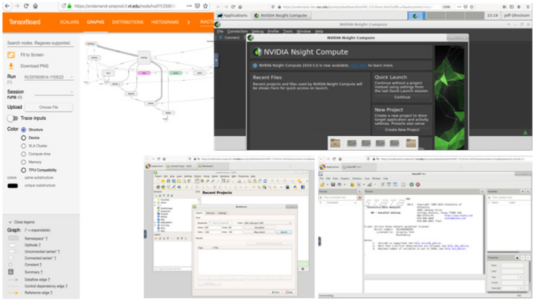

FIGURE 10 Screen shots of running apps. Left, Tensorboard running on Power system. Top right, Nsight. Bottom middle, QGIS. Bottom right, STATA.

cluster is heavily used by the Deep Learning community. OnDemand greatly simplifies startup and use of Jupyter and Tensorboard, two extremely important applications used in model building and training. While Virginia Tech uses an X86 platform to host the Open OnDemand application, other institutions are requesting native support on Power. Similarly, administrators of Nvidia DGX servers are requesting native support for Open OnDemand running on the DGX platform. Supporting any futuristic compute platform means using GPUs. Central to our strategy to increase the awareness and efficiency of our users is exporting job metrics from XDMoD to Open OnDemand. Included in the development of this capability is GPU use metrics, see top, Figure 9. Our vision is these job metrics will be prominent in the job status and summary such that users see the metrics as a diagnostic to their HPC use.

While most centers currently operate on-premises HPC clusters, many are evaluating the usage of cloud-based HPC resources, such as those available via the Google Cloud Platform (GCP). In an extension effort, we are building an extension of OOD running on the Google Cloud Platform supporting an example HPC client workflow. Specific deliverables include appropriate contributions to the OOD GitHub repos and documentation that will allow interested centers to replicate this OOD configuration on their own. While some related work has already been done, including Princeton running an instance of OOD targeted at a specific class within Amazon Web Services, and work on interfacing OOD to a Kubernetes environment, much additional work needs to be done to create a robust and production ready version of OOD that is easily deployed within a public Cloud environment as in Figure 11.

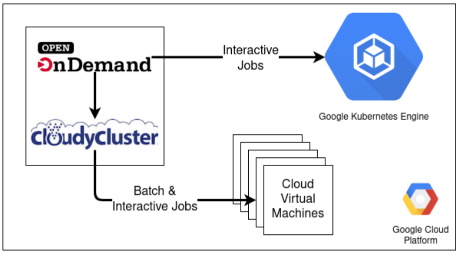

FIGURE 11 Prototype Open OnDemand deployed with CloudyCluster scheduler and integration with Google Kubernetes engine.

5.4 | Advisory Board Formation and Engagement

A major lesson learned revolves around community engagement and development towards community priorities. To facilitate collection of user/admin needs and assist in prioritizing limited resources to development priorities, we established a Scientific Community Advocacy Group (SCAG). We envision the SCAG as the voice of the community, as such it is large. To date, our SCAG consists of members from 20 different intuitions covering both industry and academia. This group meets quarterly to discuss new features, prioritization, apps development, security, and any other topic the community feels important. To more fully engage our SCAG, we are using the Quarterly Meetings to discuss development priorities, release timelines, feature requests and community expansion. For development priorities, we have adopted the following questions to help form our discussions:

1. What is the source of the feature? The proposal? The community? Both?
2. What is expected client impact?
   a. What percentage of current installed base will use the feature?
   b. Is this an new market, new platform, or expansion of existing?
   c. Is this a new installation?
3. Feasibility
   a. How much work has been done?
   b. How much work remains?
4. Do we have resources available?
5. Project sustainability (technical debt). Does it make project more stable or accessible to developers?

As we work through each of the Specific Aims in the Open OnDemand 2.0 grant, we see the the OnDemand community growing, the coverage across domains of science expanding and the overall usefulness of OnDemand making Open OnDemand an indispensable software tool for HPC use, as necessary as a browser is to the web.

acknowledgements

This work is supported by the National Science Foundation of the United States under the awards NSF SI2-SSE-1534949 and CSSI-Frameworks-1835725.

references

[1] Hudak D, Bitterman T, Carey P, Johnson D, Franz E, Bradey S, et al. OSC OnDemand: a web platform integrating access to HPC systems, web and VNC applications. XSEDE'13: Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery 2013;1(49):1–6. https://dl.acm.org/doi/10.1145/2484762.2484780.

[2] Hudak D, Johnson D, Chalker A, Nicklas J, Franz E, Dockendorf T, et al. Open OnDemand: A web-based client portal for HPC centers. Journal of Open Source Software 2018;3(25):622. https://doi.org/10.21105/joss.00622.

[3] Consortium OT. Open OnDemand: Access Clusters, Gateways and Interactive Apps 2018 9;https://figshare.com/articles/Open_OnDemand_Access_Clusters_Gateways_and_Interactive_Apps/7069691.

[4] Palmer JT, Gallo SM, Furlani TR, Jones MD, DeLeon RL, White JP, et al. Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resources. Computing in Science Engineering 2015;17(4):52–62. https://doi.org/10.1109/MCSE.2015.68.

[5] Adams A, Avila K, Basney J, Brunson D, Cowles R, Dopheide J, et al. Trusted CI Experiences in Cybersecurity and Service to Open Science. PEARC '19: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning) 2019;1(60):1–8. https://dl.acm.org/doi/10.1145/3332186.3340601.