

# Open OnDemand: Transforming Computational Science Through Omnidisciplinary Software Cyberinfrastructure

David E. Hudak, Douglas Johnson, Jeremy Nicklas, Eric Franz, Brian McMichael, Basil Gohar  
The Ohio Supercomputer Center, 1224 Kinnear Road, Columbus, OH 43212 USA  
{dhudak, djohnson, jnicklas, efranz, bmc michael, bgohar}@osc.edu

## ABSTRACT

The Open OnDemand Project is an open-source software project, based on the Ohio Supercomputer Center's proven "OSC OnDemand" platform, that enables HPC centers to install and deploy advanced web and graphical interfaces for their users. Open OnDemand also includes features such as allowing users to build their own web apps that can then be shared with other HPC users. The Open OnDemand team is completing the first year of the project and releasing its first version in the summer of 2016. In this paper, we describe the user experience and design of Open OnDemand and discuss the next steps for the project.

## CCS Concepts

• **Computing methodologies**→Simulation environments • **Information systems**→RESTful web services • **Software and its engineering**→Organizing principles for web applications • **Human-centered computing**→Web-based interaction.

## Keywords

Web platform; OnDemand; Federated Authentication; OpenID Connect; Web apps; High Performance Computing; Virtual Organizations; Cyberinfrastructure.

## 1. INTRODUCTION

The web has become the dominant access mechanism for remote compute services in every computing area except high-performance computing (HPC). Accessing HPC resources, either at the campus or national level typically requires advanced knowledge of Linux, familiarity with command-line interfaces and installation and configuration of custom client software (e.g., Secure Shell (SSH) and Virtual Network Computing (VNC)). These additional requirements create an accessibility gap for HPC. To help address this gap we have created the Open OnDemand Project, an open-source software project based on the proven Ohio Supercomputer Center (OSC) OnDemand platform [1], to allow HPC centers to provide advanced web and graphical interfaces for their users.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

XSEDE16, July 17-21, 2016, Miami, FL, USA

© 2016 ACM. ISBN 978-1-4503-4755-6/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2949550.2949644>

## 1.1 OSC OnDemand Platform

OSC OnDemand is a web platform providing users integrated access to HPC systems, web applications and VNC services hosted at OSC. A user has to know only three things: (1) the URL (<https://ondemand.osc.edu>), (2) their OSC username and (3) their password. Once authenticated, the user is presented with a set of web-based accessibility applications (or "apps") for file access, job access, and cluster access (cluster utilization and terminals). In addition, OSC OnDemand hosts AweSim AppKit-based apps, including workflow-based web apps that submit batch jobs (like many science gateways) as well as "interactive HPC" (iHPC) apps providing VNC desktop access.

OSC OnDemand has been in production since January 2013. Previous metrics taken during 2013 and 2014 show significant adoption by the OSC user community with over 800 distinct users executing over 70,000 app launches. In addition, scientists from 27 different NSF fields of science used OSC OnDemand, making it truly multidisciplinary. OSC OnDemand remains popular; in the previous year (April 2015-March 2016) it supported hundreds of users and thousands of app launches each month (see Figure 1 and Figure 2).

App Launches by Category %

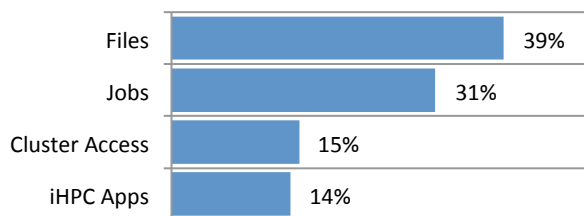


Figure 1. OSC OnDemand app launches by category for Apr 2015 – Mar 2016.

While we were gratified by the adoption of OSC OnDemand, the implementation options available in 2012 prevented the platform from fully meeting some of its goals, such as single sign-on, and "zero-install", i.e., no local SSH or VNC client installation or configuration required by the user [1]. The OSC OnDemand platform required terminal users to sign in again with their username and password and required users to install Java for the VNC client. The Open OnDemand project successfully meets these goals and introduces new features such as allowing users to build their own web apps that can then be shared with other HPC users.

Open OnDemand is the result of substantial development and integration efforts in four key areas. (1) The per-user NGINX (PUN) architecture including federated authentication using CILogon, Apache-based web proxy, per-user NGINX configuration, and UNIX pipes for secure server-side

communication between the proxy and each PUN. The PUN architecture is an original contribution of the project. (2) The file browser and file editor which, though originally based on an existing open source project, have been extensively modified. (3) The terminal, created by integrating an existing open source project with minimal effort. (4) Accessibility Apps (Dashboard, Job Constructor, Job Status, System Status, VDI and iHPC apps) built using the Rails-based AweSim AppKit (which was developed by this team on a previous project and leveraged here without minor modifications). The AweSim AppKit allows for the development of both workflow and interactive applications and includes mechanisms for user-based app creation, app sharing and app publishing. The AppKit technology is included as part of the Open OnDemand project.

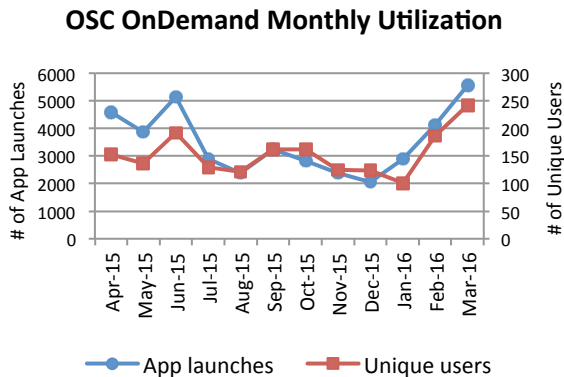


Figure 2. OSC OnDemand monthly utilization chart for Apr 2015 – March 2016.

## 1.2 Open OnDemand Project

A number of HPC centers have requested access to OSC OnDemand’s source code to provide OnDemand platforms for their users. In addition, we feel that OnDemand’s unique architecture has the potential to transform cyberinfrastructure by providing a platform to enable a new delivery method for scientific web tools like HubZero apps, XSEDE Science Gateways, Jupyter notebooks and workflow tools like Pegasus. In response to these opportunities, we developed the Open OnDemand Project, an open source platform for HPC and remote computing access with the following goals: (1) Generalize the infrastructure of OSC OnDemand to improve portability, robustness and security thus creating a community-developed open source software package called Open OnDemand. (2) Improve the interface capabilities of Open OnDemand by updating and expanding the accessibility apps, including integration of NSF-funded Globus Online [2] and Open XDMoD projects [3]. (3) Conduct a program to engage departmental, campus and national HPC users and administrators on enhancing HPC inclusivity by enhancing OSC’s existing OnDemand documentation and training materials. (4) Leverage Open OnDemand as a platform to support existing web-based applications such as XSEDE Science Gateways and HubZero applications.

## 2. RELATED WORK

There has been a large and impressive body of work creating grid services and science gateways. These projects have been compared to OSC OnDemand in a previous paper [1] and needn’t be repeated here. As an open-source software project hoping for

adoption at supercomputer centers, the Open OnDemand project shares similar organizational goals as the Open XDMoD project [3] and HubZero [4]. Open OnDemand’s job monitor and system monitor apps overlap in function with Open XDMoD but were included to provide a useful platform for Open OnDemand users. Also, the development team felt that, given the ease of creating these apps in the AppKit infrastructure, why not make them? That said, tighter feature integration and reducing duplication between Open OnDemand and Open XDMoD is a goal for year 3 of the project. Open OnDemand’s app creation and sharing are similar in spirit and owe a debt to HubZero [4], but differ in implementation (for example, Open OnDemand’s per-user Nginx architecture). Also, HubZero provides web interfaces for creating, submitting, reviewing and sharing new apps. Open OnDemand currently handles these tasks in a manual fashion using command-line tools and are not at this time as mature or user-friendly as HubZero. In the future, Open OnDemand will incorporate similar features depending on the interest in user-contributed AppKit apps.

## 3. OPEN ONDEMAND USER EXPERIENCE

### 3.1 Accessibility Apps

#### 3.1.1 Dashboard

After logging in, the user is first presented with the Dashboard app. The Dashboard is a navigational aid to launch apps, whether they are the accessibility apps or user shared apps (see Figure 3). The user is presented with a set of menus used to navigate to the system-installed accessibility apps, apps they are currently developing, and apps being shared with them by other users. Because the NGINX process serving the Dashboard is running as the user, app sharing is maintained through the use of system-level file permissions granted to the user by the app developer. Giving a user access to an app is as easy as giving that user permissions to the path of the app on the shared file system. The app will immediately appear in the Dashboard for that user upon a refresh of the web page.

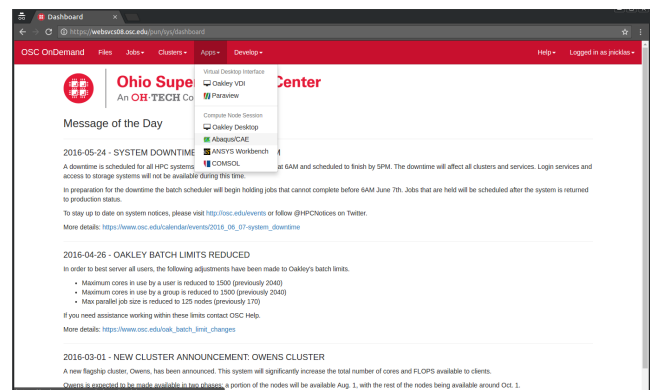


Figure 3. Open OnDemand Dashboard app.

#### 3.1.2 Files

At most HPC Centers users are given a home directory that is accessible from all systems. Open OnDemand has incorporated the open source Cloud Commander project (<http://cloudcmd.io>), with heavy modifications, into the File Explorer app (see Figure 4) to provide an interface to not only the user’s home directory but also to all the shared file systems provided by the HPC center. The File Explorer app is a Node.js application that allows the user to browse directories, rename files, copy files, move files,

compress files and directories, as well as upload and download files. Since the NGINX instance serving this web application runs as the user, system-level file permissions are enforced by the server and users cannot read or write to files unless they have the proper system-level permissions to do so.

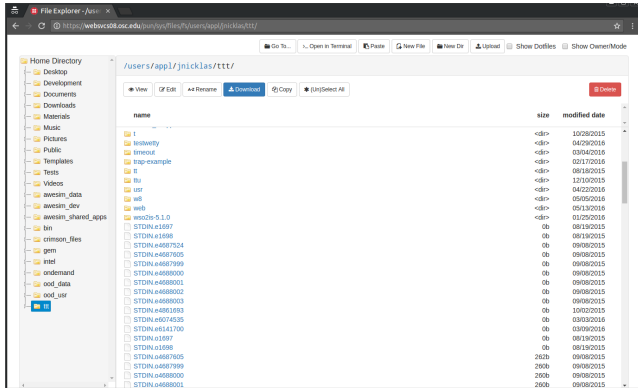


Figure 4. Open OnDemand File Explorer app.

While the File Explorer provides an intuitive user interface for uploading and downloading files, the underlying hardware and web server configuration hosting the app limits the speed and capacity of transfers. In recent tests we demonstrated successful uploads of up to 10 GB limited only by local disk space on the server. For uploads an order of magnitude larger we plan to incorporate Globus Online as OnDemand’s preferred large-scale data transfer solution in year 3 of the project.

OnDemand provides a File Editor app (see Figure 5) for manipulation of text files directly on the file system. This editor app relies on the open source Ace Editor (<https://ace.c9.io>) which is an embeddable code editor written in Javascript. The editor app allows the user to modify font size, color schemes, interface modes, and syntax highlighting when editing a file. It also provides undo/redo capabilities as well as a separate “Save” button so that any uncommitted changes must be saved explicitly by the user. Similar to the File Browser, the user will only be able to edit files they have read and write permissions to on the local file system. The editor app isn’t directly launched from the Dashboard, but instead is launched from within the File Explorer app when the user chooses a text file to edit.

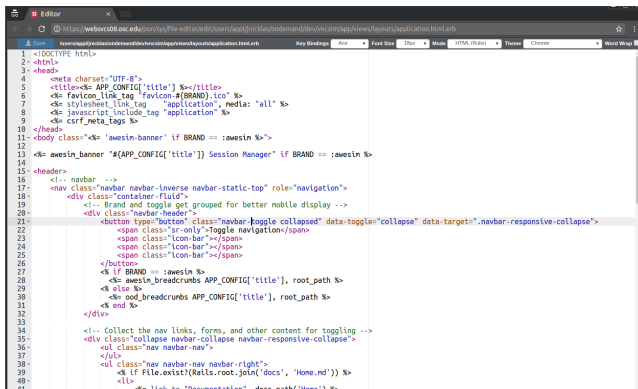


Figure 5. Open OnDemand File Editor app.

### 3.1.3 Jobs

OnDemand provides an alternative to the command line for submitting, deleting, and viewing the status of jobs with the incorporation of the Job Status and Job Constructor apps.

The Job Status app displays a table of jobs currently active on the various batch servers provisioned by the HPC center. A user can filter the list of jobs based on jobs he or she owns, as well as the jobs submitted by users in his or her primary Unix group. The Job Status app also provides flexible search capabilities, which allow the user to search across: job id, job name, user name, group ownership, status of job, and cluster. When the user selects a job from the list, the details of the job are displayed including resources requested by job, resources used by job, owner, charged account, and status of job. For running jobs, performance reports generated by Ganglia are available for all the nodes in the time span that the job has been running. These reports can be useful to the user for debugging and profiling purposes. The Job Status app also provides an intuitive interface that allows the owner of the currently selected job to delete the given job, jump to the output directory for the given job in the File Explorer app for viewing/editing results or logs, or launch the Terminal app with the current working directory set to the job’s output path. This integration between the various system accessibility apps hopes to provide a fluid job management experience for the users.

The Job Constructor app is a for batch job creation and control (see Figure 6). It provides a web interface for the creation, modification, submission, monitoring, cancellation and deletion of batch jobs. In our experience, most batch jobs are created by copying and customizing existing “known good” jobs. The Job Constructor app workflow is built around known good jobs called “templates”. Templates are directories that contain the skeleton of a running batch job. These can be default templates supplied by the Job Constructor itself, old batch jobs completed by the user, or even shared directories between colleagues and advisors. The user creates a new job from a template and then modifies the parameters to meet the goals of the simulation. The user can then submit the job to any available batch server and monitor the status of the job in real-time from within the app. When the batch job simulation has completed, the app provides a direct link to the working directory in the File Access app so that the user can view or download the output files. Job Constructor also allows the user to copy the completed job for further post-processing or restarting, and it also provides support to delete jobs and their directories to free up disk space as necessary.

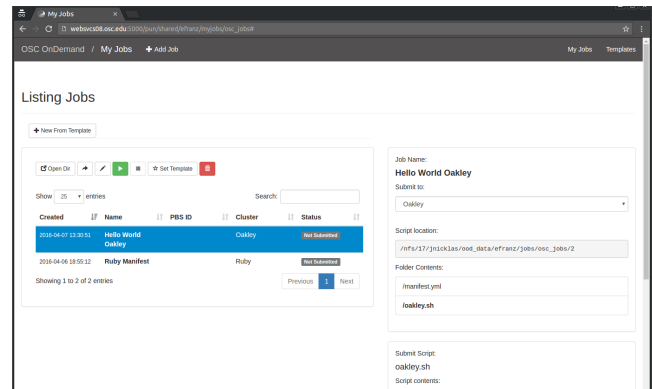


Figure 6. Open OnDemand Job Constructor app.

### 3.1.4 Cluster Access

Being able to view the resource utilization of a current cluster as well as accessing a given cluster is handled by the included System Status and Terminal apps. The System Status app gives the user overview performance reports for the individual login and compute nodes of the clusters supported by the HPC center. These reports include CPU, memory, and network utilization.

The System Status app leverages the open source Ganglia Monitoring System [5] for monitoring and generation of the reports provided within the web app. However, we do not want the functionality limited to Ganglia and plan to engage the community to support multiple monitoring solutions.

One of the goals of the Open OnDemand project is to reduce reliance on a command line interface. This can be seen in the case of the Job Status app replacing the behavior of the `qstat` command line tool as well as the Job Constructor app replacing the `qsub` command line tool. However, we feel it's unlikely to fully replace the flexibility provided by a command line interface, so a Terminal App is included in OnDemand. The Terminal app (see Figure 7) is a customization of the open source Wetty app (<https://github.com/krishnasrinivas/wetty>). The Wetty app is a Node.js application taking advantage of the ChromeOS HTML terminal emulator `hterm` written entirely in Javascript. It uses the WebSocket protocol for enhanced response time and reduced overhead compared to using Ajax polling. Users can create as many terminal sessions as they want in their browser using separate tabs. They can also choose to connect their terminal to any of the available cluster login nodes. The unique deployment of the per-user Nginx (PUN) stack ensures that the terminal session runs as the appropriate system-level user that is authenticated through Open OnDemand.

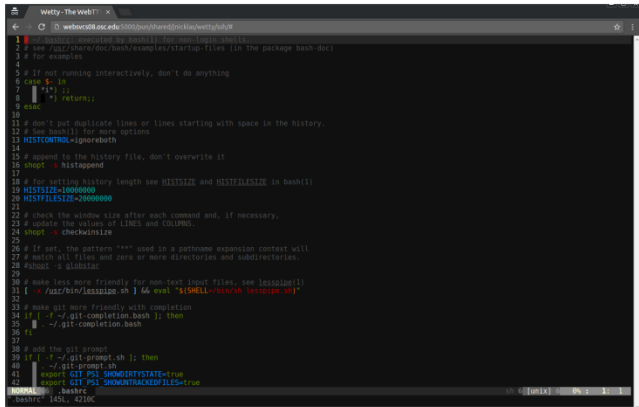


Figure 7. Open OnDemand access to Terminal app.

### 3.1.5 Interactive HPC (iHPC) Apps

Regardless of the means employed to produce results, users typically want to view their results graphically. OnDemand supports this through what is termed “interactive HPC” (iHPC) Apps. A common scenario at HPC centers involves users launching GUI applications through a remote VNC session. OnDemand provides access through a web interface to VNC applications running on compute nodes launched from within a batch job. Users are able to launch and access remote desktops (see Figure 8), for applications such as Abaqus, Ansys, COMSOL, and Paraview within a VNC session hosted on one or more compute nodes. In order to provide a VNC connection without requiring the user to download and install specialized VNC client software, Open OnDemand utilizes the open source HTML5 VNC client noVNC (<https://kanaka.github.io/noVNC/>). This VNC client runs completely in any modern browser including mobile browsers and takes advantage of the secure WebSocket protocol keeping all VNC traffic encrypted and tunneled through the firewall-friendly port 443. The connection is securely established between the user’s browser and the backend VNC server running on the compute node by means of single-use tokens. This eliminates the possibility of a malicious user at the

HPC center directly connecting to the VNC server on the compute node and hijacking the session.

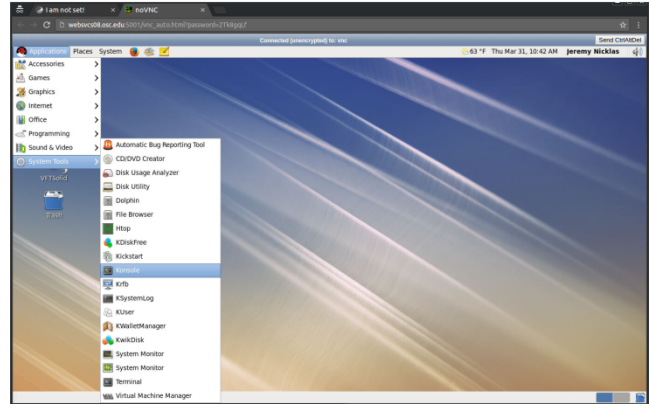


Figure 8. Open OnDemand access to iHPC apps. This figure shows a remote desktop on a compute node displayed in the noVNC client.

Open OnDemand also provides a secure web proxy to access backend web servers running off of compute nodes launched through a batch job. This allows a user to run modern servers such as a Jupyter notebook server, RStudio Server, or COMSOL Server within a batch job. The user then is able to take advantage of the reverse proxy to connect to their server securely through the OnDemand portal in their browser. There is no need for a separate VPN or SSH tunnel to establish the connection. Care must be taken by the developer that a proper authentication module is in place for the chosen backend web server to prohibit local connections from unauthorized users.

## 3.2 Creating Apps

Web applications can be written using the Ruby, Python, or Node.js (Javascript) languages within the developer’s home directory. Each of these languages provides a multitude of web frameworks to choose from when creating an app. The Open OnDemand project in particular the AweSim AppKit built on the Ruby on Rails web framework (<http://rubyonrails.org>). The AweSim AppKit (see Section 4.2.1) eases development of apps that interface with HPC resources (e.g., batch server and shared file system). With the AppKit, a developer can quickly and easily scaffold an entire web app using just a batch script template as a reference.

The scaffolding tools provided in the AppKit allow the developer to quickly generate a straightforward fully-functional web app. This generated bare-bones web app presents users with a simplified web form that displays the parameters outlined in the batch script template that the user fills out to generate batch jobs. The users can then submit, view, and control these jobs in a searchable and sortable dynamic table. Building web apps around commonly used solvers will decrease the cost of training new users and students in both HPC usage as well as the complex command-line workflows required by most solvers.

## 3.3 Sharing Apps

The Dashboard app is served by an NGINX process running as the authenticated user and programmatically checks if it has the necessary system-level file permissions to access any of the available apps. If the user does have permissions to a given app, a link followed by the description for the given app is displayed to



the user. The user can then click on the link and the app will launch.

If an app that the user wants to access is not displayed on the Dashboard, the user can make a request to the developer for access to the app. The developer will only need to add the appropriate permissions to the app directory to grant access to the given app for the user. The specified app will now appear in the Dashboard upon a browser refresh by the user.

## 4. OPEN ONDEMAND DESIGN

### 4.1 Infrastructure

#### 4.1.1 Federated Authentication

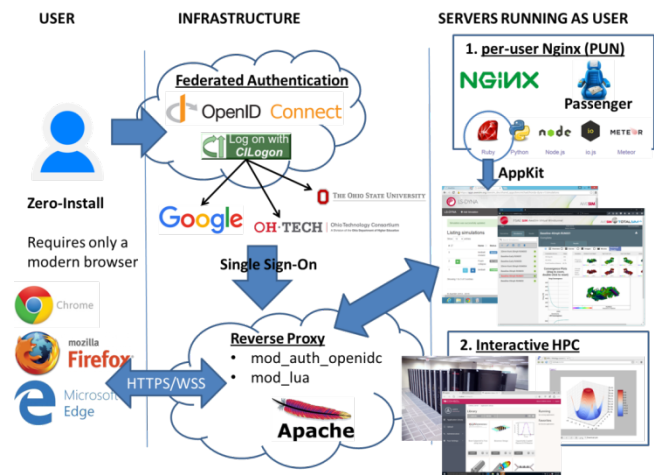
The first component any user sees when accessing Open OnDemand is the authentication portal. The goal of the authentication portal is to not only authenticate the user against a trusted identity provider but also provide the user a single sign-on experience. The single sign-on experience allows the user to sign in with a single username and password and seamlessly gain access to all of the applications provided by the Open OnDemand portal. The authenticated user token can then be provided to the individual apps for further identification as well as authorization purposes.

Open OnDemand uses two such authentication mechanisms: OpenID Connect (<http://openid.net/connect/>) and the CILogon Service [6]. OpenID Connect is an open source identity layer built on top of the OAuth 2.0 protocol. It is deployed and in use by many leaders in the industry including Google, Microsoft, Ping Identity, and PayPal [7]. An OpenID Connect identity provider server is deployed that is integrated with OSC's LDAP server for local user authentication. This allows users to authenticate with their OSC username and password.

CILogon is an open source project funded by the NSF and the DOE. CILogon allows users to use InCommon (SAML) or OpenID identity providers for external authentication with their home organization. With the addition of CILogon, an authenticated user can register multiple identities that are subsequently mapped back to their local HPC account username. The goal is for OSC users to authenticate using their university academic credentials when accessing OSC resources.

#### 4.1.2 Reverse Proxy

After successful authentication the user is redirected to the front facing Apache reverse proxy. The proxy's responsibilities include (1) facilitating single sign-on support, (2) tunneling all web traffic through port 443 to better support firewalls, and (3) encrypting all traffic using the standard SSL protocol (see Figure 9).



**Figure 9. Diagram of the Open OnDemand Infrastructure workflow.**

The Open OnDemand project uses the Apache HTTP Server 'httpd' due to its mature and flexible module support. In particular, the `mod_auth_openidc` module ([https://github.com/pingidentity/mod\\_auth\\_openidc](https://github.com/pingidentity/mod_auth_openidc)) is a well-established authentication/authorization module that authenticates users against an OpenID Connect Provider. This module enables the Apache reverse proxy to operate as an OpenID Connect relying party for any OpenID Connect identity provider.

The `mod_lua` module (<http://www.modlua.org>) is another crucial component in the underlying Open OnDemand httpd infrastructure. The Open OnDemand project leverages `mod_lua` in the reverse proxy for (1) mapping the authenticated username to the system-level username, (2) generating, maintaining Nginx configs, and controlling the per-user Nginx (PUN) instances discussed later, (3) proxying traffic from the corresponding user to their respective PUN process, as well as (4) proxying traffic from the user to the requested interactive HPC app running within the local network (e.g., a Jupyter notebook web server running on a compute node). The latter functionality precludes the need for a VPN or ssh tunnel which is commonly used at HPC centers.

For controlling and maintaining the PUN processes, the Lua portion of the reverse proxy makes system calls to a Ruby script called `nginx_stage` with sudo privileges. This allows the reverse proxy running as the 'apache' user to launch a PUN server as the authenticated user. This Ruby script is responsible for generating the necessary Nginx configuration files from a set of templates, launching the PUN as the specified user, and sending Unix signals to stop or restart the appropriate PUNs.

#### 4.1.3 Per-user Nginx Servers

The proxy parses the user-specified link and maps it to the corresponding resource. For the Dashboard as well as the other accessibility apps aside from the iHPC apps the user's traffic will be proxied to a backend per-user Nginx (PUN) server. The PUN is defined as an Nginx server running as the system-level user that listens for connections on a Unix domain socket.

Running a web server as the user, dubbed per-user web server (PUWS), was first successfully explored in the original OSC OnDemand project [1]. It provides for proper accounting of user-requested resources on the system as well as enforcing system-level security through the use of Unix permissions. At worst, if a malicious user compromises a web application they would be limited to the files and privileges of the compromised account.

The PUN takes advantage of the underlying file permissions by listening for connections on a Unix domain socket. Only the reverse proxy user will have permissions to communicate with the PUN preventing the threat of a user on the localhost directly connecting to another user's PUN. Using a Unix domain socket also has the added benefit that the proxy can easily map an authenticated user to a respective socket file. This mapping can otherwise be overly complex using a finite number of local TCP ports if using IP sockets.

Phusion Passenger (<https://www.phusionpassenger.com>) is integrated into the PUN to further ease the web application development process. Phusion Passenger is an application server that supports Ruby, Python, and Node.js web application development. Using a web framework relying on one of these languages can significantly reduce the time and expense needed to build and deploy a user-facing web application.

## 4.2 App Architecture

### 4.2.1 AppKit

The Open OnDemand portal leverages the AweSim AppKit, a development environment previously built by the Open OnDemand team, which will be included in the Open OnDemand release. AppKit makes extensive use of the Ruby language in its web applications that interact with underlying HPC resources. The AppKit consists of a collection of Ruby libraries that provides an interface for Ruby web applications to interact with the local batch server as well as shared file system. The AppKit provides an easy-to-use API for creating multi-job workflows on the backend batch server. This allows for the development of apps that submit sequential workflows for pre- and post-processing as well as parallel workflows that facilitate parameter sweeps.

### 4.2.2 App Sharing

The Open OnDemand project provides the flexibility for apps to be shared amongst HPC users. Web applications can be developed and deployed by HPC users in their own home directories. The Open OnDemand project is designed with both a sandbox environment for the private development of an app by a user as well as a shared environment akin to a production environment where the app is shared with other HPC users. The shared apps can be launched by other users in their respective PUNs when they visit the appropriate shared app URL.

Fine-grained authorization is maintained by the app developer through the use of POSIX ACL file permissions giving individual users and groups access to any given app. A more global and system-controlled authorization is maintained by the system administrator, e.g., allowing only users in the same primary group the ability to share apps with one another.

### 4.2.3 App Publishing

The Open OnDemand project can be easily configured to host apps from any location on the file system. This makes app publishing for system administrators or a system-user developer as easy as copying an app directory into the required file system location and performing any necessary setup procedures for the app. The app will then appear in the Dashboard for users with the appropriate permissions to access the app directory. A description of the app will also appear on the Dashboard if a manifest file was supplied by the developer in the app directory. The user then successfully initiates and launches the app under his or her PUN environment after navigating to the appropriate URL from within the Dashboard.

## 5. PROJECT PLAN

Open OnDemand is an open source project released under the MIT License. The project contains the Open OnDemand platform (Reverse Proxy, Per-user Nginx Web Server, App Kit, and App Sharing/Publishing). The dependencies for the bundled accessibility apps (Cloud Commander, Wetty and noVNC) are open source as well, but with varying licenses. To measure community adoption, the Open OnDemand team will track and report on the following metrics: total active Open OnDemand sites, total distinct users, and total launches per app. Of course, as an open source project, Open OnDemand must rely on community support for collecting these metrics. With the release of the first beta, engagement efforts are beginning including creating system administrator documentation, online discussion forums and training materials based on existing OSC OnDemand materials.

## 6. CONCLUSION

The Open OnDemand project hopes to improve the accessibility of remote computational resources and improve the productivity of scientists and researchers by achieving the following goals. (1) Open OnDemand will bring broader accessibility to national cyberinfrastructure services. Scientists using Open OnDemand at campus, regional and national HPC centers will be able to use web and graphical tools for their work, versus the traditional command line based access techniques. (2) Open OnDemand will increase integration of national cyberinfrastructure software. Open OnDemand will interoperate with Globus Online, Open XDMoD and SUPREMM to provide scientists with a unified set of tools for using and monitoring cyberinfrastructure. (3) Open OnDemand will lead to a proliferation of customized accessibility apps for science. Researchers could use the Open OnDemand platform to serve up previously developed web apps (like Galaxy or an XSEDE Science Gateway) or to develop their own apps from scratch. This allows them to make specialized domain-specific workflows and expertise available to more collaborators and users, both from academia and industry.

## 7. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation of the United States under the award NSF SI2-SSE-1534949. The authors would like to thank the reviewers for their comments, which considerably helped strengthen this paper.

## 8. REFERENCES

- [1] Hudak, D. E., Bitterman, T., Carey, P., Johnson, D., Franz, E., Brady, S., and Diwan, P. 2013. OSC OnDemand: a web platform integrating access to HPC systems, web and VNC applications. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*. (San Diego, CA, USA, July 22 - 25, 2013). XSEDE '13. ACM, New York, NY, 49, DOI=<http://dx.doi.org/10.1145/2484762.2484780>.
- [2] Foster, I. Globus Online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing* 15, 3 (May 2011), 70-73. DOI=<http://dx.doi.org/10.1109/MIC.2011.64>.
- [3] Palmer, J., Gallo, S., Furlani, T., Jones, M., DeLeon, R., White, J., Simakov, N., Patra, A., Spherhac, J., Yearke, T., Rathsam, R., Innus, M., Cornelius, C., Browne, J., Barth, W., and Evans, R. Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resources. *Computing in Science & Engineering* 17, 4 (May 2015), 52-62. DOI=<http://dx.doi.org/10.1109/MCSE.2015.68>.

- [4] McLennan, M., and Kennell, R. HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering. *Computing in Science and Engineering* 12, 2 (March 2010), 48-52. DOI= <http://dx.doi.org/10.1109/MCSE.2010.41>.
- [5] Massie, M. L., Chun, B. N., and Culler, D. E. The ganglia distributed monitoring system: design, implementation and experience. *Parallel Computing* 30, 7 (July 2004), 817-840. DOI= <http://dx.doi.org/10.1016/j.parco.2004.04.001>.
- [6] Basney, J., Fleury, T., and Gaynor, J. CILogon: A Federated X.509 Certification Authority for CyberInfrastructure Logon. *Concurrency and Computation: Practice and Experience* 26, 13 (September 2014), 2225-2239. DOI= <http://dx.doi.org/10.1002/cpe.3265>.
- [7] Thibeau, D. The OpenID Foundation Launches OpenID Connect Certification Program, 2015. Retrieved March 30, 2016, from OpenID: <http://openid.net/2015/04/17/openid-connect-certification-program/>.