Comparing Open Ondemand and Jupyterhub as Interactive HPC Gateways

Michael Milligan
Graham T. Allan
Nicholas J. H. Dunn
Matt Mix
milligan@umn.edu
gta@umn.edu
dunn0404@umn.edu
mattmix@umn.edu
University of Minnesota
Minneapolis, Minnesota, USA

ABSTRACT

The Minnesota Supercomputing Institute (MSI) at the University of Minnesota has adopted a goal of supporting Interactive high performance computing (HPC) as a first class service. For several years MSI has used Jupyterhub to provide a web gateway to interactive Jupyter notebook environments on HPC resources. In the past year MSI has additionally adopted Open OnDemand as a general purpose web gateway for interactive HPC services. This places MSI in the position to provide an experience-informed comparison of two popular open-source platforms for interactive web-based access to HPC resources.

Open OnDemand is designed specifically to provide access to HPC resources, and includes facilities to manage files and jobs, access in-browser command line sessions, and provide virtual desktop access to GUI applications as well as proxied access to web applications such as Jupyter servers. Jupyterhub, in contrast, is designed to provide multi-user access to Jupyter servers, and when configured with the Batchspawner extension is able to manage such servers on scheduled HPC cluster resources. Jupyterlab, when accessed through a Jupyterhub so configured, likewise offers access to in-browser facilities such as command line terminals and file management, and can proxy access to some other web applications, in addition to supporting the main Jupyter Notebook application. From a systems perspective, both platforms provide a highly configurable base on which to build customized gateway solutions, but the differing focus of the two projects results in different characteristics and challenges.

Despite different origins and project focus, Jupyterhub and Open OnDemand have evolved to be able to provide strongly overlapping services to users of an HPC center. This poster will compare the characteristics of the two platforms and the different advantages and limitations of each, from the perspective of an HPC center that has deployed both into production for use by a large base of research users.

CCS CONCEPTS

• **Human-centered computing → Collaborative and social computing systems and tools**; • **Social and professional topics** → *Software selection and adaptation*; • **Software and its engineering** → *Software usability*.

KEYWORDS

interactive computing; HPC; science gateways; Jupyter; Jupyterhub; Open OnDemand

ACM Reference Format:
Michael Milligan, Graham T. Allan, Nicholas J. H. Dunn, and Matt Mix. 2023. Comparing Open Ondemand and Jupyterhub as Interactive HPC Gateways. In *Practice and Experience in Advanced Research Computing (PEARC '23), July 23–27, 2023, Portland, OR, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3569951.3597602

1 INTRODUCTION

High performance computing has traditionally centered on batch computing, but computational researchers increasingly demand interactive access modes. Interactive HPC allows users to interact with the system in real-time, explore data interactively, and iterate their code and workflows in real time. In a parallel development, the web has rapidly become the dominant access paradigm for remote access to computational resources of all kinds; while HPC as a field lagged this trend, today's researchers increasingly expect web-based access to these systems as well. The Minnesota Supercomputing Institute aims to support interactive HPC as a first class service, and increasingly does so using web technologies[3].

With the adoption of the Jupyterhub/Batchspawner platform, MSI's interactive HPC offerings fell into three primary categories: command line interfaces accessed via SSH, remote desktops provided by NoMachine NX and NICE-DCV EnginFrame, and Jupyter[1] sessions accessed via Jupyterhub. This state of affairs presented both operational and user support challenges. In particular, maintaining two remote desktop portals, which each required a special-purpose compute resource separate from MSI's main compute clusters, proved costly in staff time and licensing costs. We sought a

PEARC '23, July 23–27, 2023, Portland, OR, USA

ACM ISBN 978-1-4503-9985-2/23/07.
https://doi.org/10.1145/3569951.3597602

solution that would simplify operations while giving users a superior experience in which they could directly access compute cluster resources. Open OnDemand[2] (OOD) came to our attention as a promising open source solution that could provide this service using web technology, and a number of other interactive services as well. MSI deployed OOD for general use in December 2022 and is in the process of retiring the older remote desktop service offerings.

This presents the opportunity to perform a direct comparison. Aside from traditional SSH shell sessions, MSI now supports two major interactive HPC technologies: Jupyterhub and OOD. Other than remote desktop access, these technologies support a strikingly overlapping set of interactive services. At first glance, this may be surprising. OOD is purpose-built to provide interactive services in HPC environments. Jupyterhub, in contrast, exists to enable the use of Jupyter in multiuser environments. Because of the wide diversity of such environments, Jupyterhub is built on a highly modular architecture. Starting in 2015, MSI supported the development of Batchspawner, a Jupyterhub spawner plugin that enables it to launch Jupyter sessions in scheduled (Slurm/PBS/Condor/etc) environments. Meanwhile, Jupyter means more than just Jupyter Notebooks. The Jupyter web application has long provided command line terminal and file browsing capabilities, and the extensible Jupyterlab application has further developed these possibilities. As a result, when configured for interactive HPC use, Jupyterhub and OOD share a notably similar architecture: both consist of a web application able to serve a UI, submit and manage jobs on behalf of authenticated users, and create proxied web connections to services started within those jobs on compute resources.

2 COMPARISON: DEPLOYMENT AND MANAGEMENT

ability to deploy and manage them via puppet modules. Jupyterhub consists of a python application and accompanying web proxy and database services, and can be installed in a number of ways. MSI chose to develop a puppet module that deploys Jupyterhub, which occurred alongside the development effort that supported the creation of Batchspawner.

Open OnDemand's web service was easy to deploy through the OSU provided puppet module, which offered flexible configuration for most options. MSI did need to develop patches to configure some missing desired options, and OSU promptly accepted these patches. The OOD project provides well-written documentation that demonstrated what configurations could be set to meet our needs.

Both OOD and Jupyterhub can utilize properly configured external software, whether provided by the HPC center or installed by the user. For an HPC center like MSI that aims to provide a large library of user-facing software, this has the consequence that the provided software must be made available in a way that can be used via the gateways. Naturally a large fraction of the software library consists of command line applications, and these can be used as usual from a terminal environment on either platform. Another major class of interactive application either provides or requires the use of a GUI. The flexibility of the virtual desktop environment that OOD provides means that any module that provides a graphical interface could be tested and configured. Through its capability for GPU-accelerated virtual desktops, OOD was able to displace two earlier remote desktop technologies (NICE-DCV and NX), and simplify our catalog of services. It should be noted that efforts do exist in the Jupyter ecosystem to support remote desktop sessions via websocket NoVNC, the same underlying remote display technology as OOD uses. While MSI has used these extensions in other contexts, we have not successfully deployed this as a general compute cluster remote desktop service, and thus we do not evaluate that approach here. In both cases, the loose coupling between software applications and the gateway means that they can be updated largely independently of each other. OOD does however provide the capability to launch directly into graphical applications via AppKit 'apps', and this does create a dependency that must be tested for the apps so installed. Currently MSI provides such convenience apps in OOD for ANSYS Workbench, IGV, Matlab, Abaqus, COMSOL, IDL, and Mathematica, but supports over 500 software modules that receive no special handling for OOD.

Open OnDemand provides broad freedom in how AppKit 'apps' are started on compute nodes, giving full control to the administrator to choose which way works best for any individual app: managed environment modules, local installation, or through containers. Additionally, the ability to use apps developed in a user's home directory allows quick iteration when developing or customizing new apps. With Jupyter, there are more constrained mechanisms for launching additional applications in the compute environment: under user control from a Notebook or CLI session, or via Jupyterlab plugins through the launcher. In general, applications launched in this way must either be Jupyter language kernels (such as python, R, etc) or must be accessible as a proxied web application via the jupyter-server-proxy mechanism. As mentioned above, this could in principle but currently does not at MSI include graphical applications.

3 COMPARISON: USER SERVICE

OOD presents more complexity (options) to the user than Jupyterhub, and as a result needs more documentation and user support to get new users up and running. Once they are familiar with the basics, however, OOD quickly opens up a wide variety of interactive workflows via the virtual desktop and the ability to flexibly choose the resources for the submitted jobs. The virtual desktop provides users with a familiar graphical interface where they can perform activities like file management, document viewing, and coding without needing to learn the command line tools for these tasks. This makes OOD a flexible on-ramp for novice users who may be intimidated by the command line interface. Additionally, because of the AppKit ecosystem, frequently requested features such as graphical virtual desktops and proxied RStudio Server sessions are easily supported via OOD. In contrast, as configured, our Jupyterhub instance provides less flexibility in the resources that can be requested for a job, so users need only select the type of job they would like from a short list before they can be up and running in an interactive Jupyterlab environment, with the corresponding interactive HPC capabilities. However these capabilities do not include some features, such as graphical desktops and RStudio, that Jupyter can technically provide but MSI was unable to adequately support, perhaps because the Jupyter ecosystem is less focused on HPC environments.

For users wanting an interactive python or R environment, Jupyterhub provides a faster way to launch a session. Once in the Jupyterlab environment, similar file management, editing, and command line tools are available. However, it poses a user education challenge to direct users who need, e.g. a file browser, to an unrelated application that they may know, if at all, as a python or data science tool. However, Jupyter notebooks and Jupyterlab are widely used outside of the HPC community and have a significant amount of tutorial materials available for free online. So users are more likely to show up already having experience with Jupyter, and will have an easier time finding external documentation to fill in the gaps in their knowledge.

4 FUTURE WORK

In the coming months we will migrate to the latest release of OOD, which will be our first experience navigating a major version increment. We also plan to integrate XDMoD, which will give OOD the additional capability to present metrics information. In the medium term we continue to support Jupyterhub as well; as MSI continues to develop cloud-integrated workflows, we plan to explore Jupyterhub's capabilities to spawn sessions via a wide variety of technologies beyond cluster scheduler jobs. As our users gain more experience with OOD we hope to gather comparative user experience data as well.

REFERENCES

[1] Brian E. Granger and Fernando Pérez. 2021. Jupyter: Thinking and Storytelling With Code and Data. *Computing in Science & Engineering* 23, 2 (March 2021), 7–14. https://doi.org/10.1109/MCSE.2021.3059263

[2] 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 (2018), 622. https://doi.org/10.21105/joss.00622

[3] Michael B. Milligan. 2018. Jupyter as Common Technology Platform for Interactive HPC Services. In *Proceedings of the Practice and Experience on Advanced Research Computing* (Pittsburgh, PA, USA) *(PEARC '18)*. Association for Computing Machinery, New York, NY, USA, Article 17, 6 pages. https://doi.org/10.1145/3219104.3219162