

Accessing Distributed Jupyter/Spark in OnDemand



Jeremy W. Nicklas, Eric Franz, Alan Chalker, Doug Johnson,
Morgan Rodgers, David E. Hudak
Ohio Supercomputer Center

Come see
my poster tonight
for more info!

OPEN

OnDemand

Overview

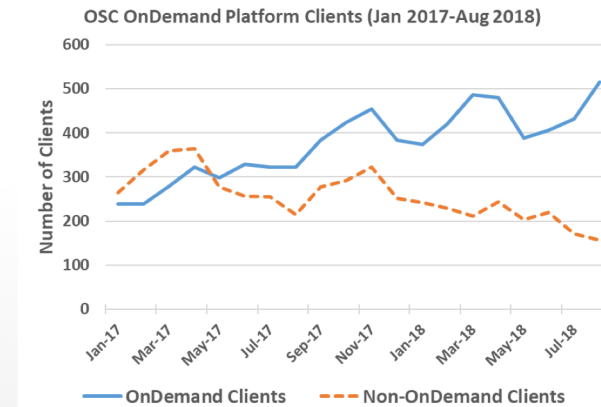
- Easy to install and use
- Web-based access to supercomputers
- Support for interactive supercomputing

Features include

- Plugin-free web experience
- Easy file management
- Command-line shell access
- Job management and monitoring
- Graphical desktops and applications

OSC Install Details and Impact

- Launched Sep. 2016, serving OSC clients globally
- % of users has steadily increased since launch

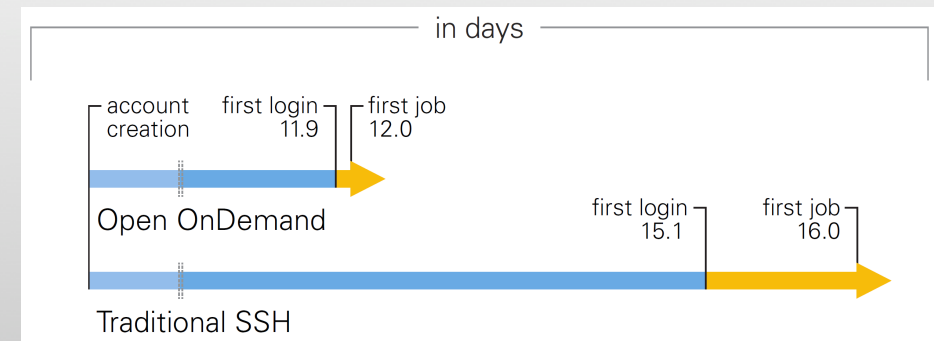


- **Improving Time to Science:** new OnDemand users start faster than ssh users: first login and first job



Ohio Supercomputer Center

An **OH-TECH** Consortium Member

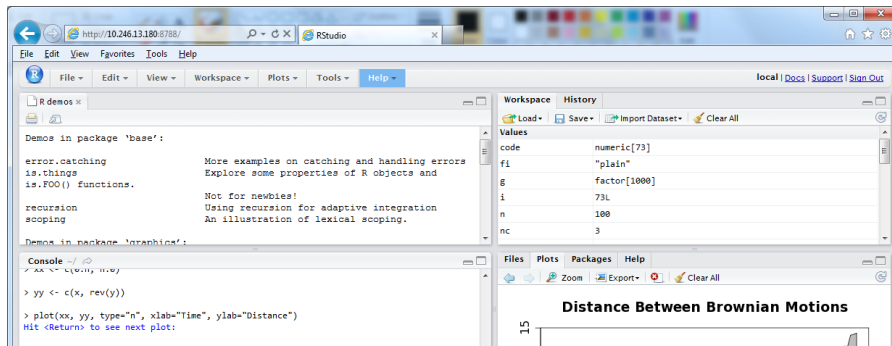


Interactive Apps

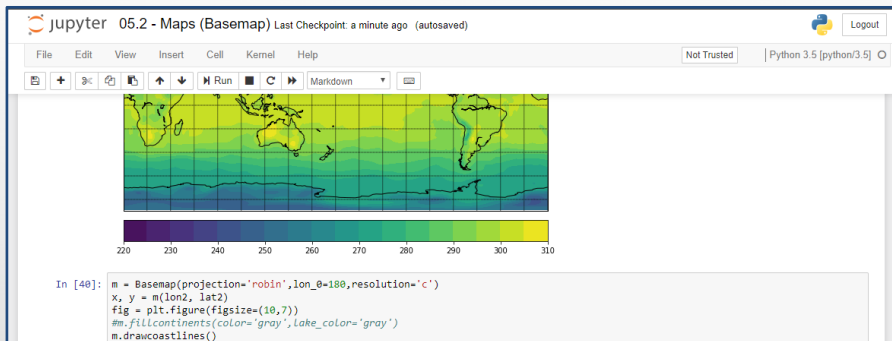
&

Cluster Access

RStudio Server – R IDE



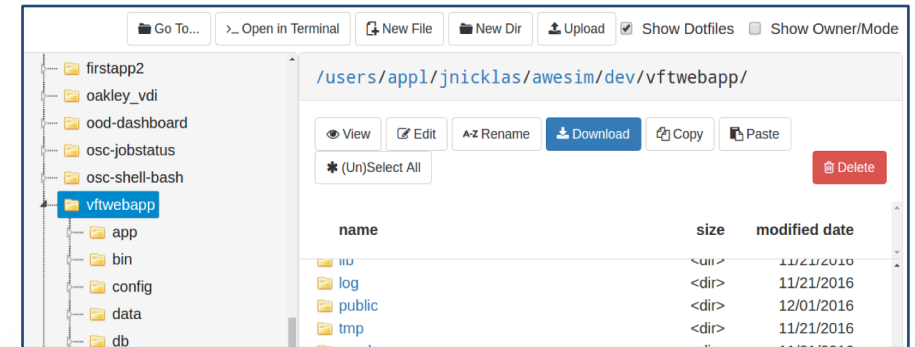
Jupyter Notebook – Python IDE



Come see
my poster tonight
for more info!

And many more, such as ANSYS Workbench, Abaqus/CAE, MATLAB, Paraview, COMSOL Multiphysics

File Access (browse, edit, etc)



Manage Jobs (view, submit, etc)

Active Jobs									
ID	Name	User	Account	Time Used	Queue	Status	Cluster		
> 3057900.owe...	high_yp_PIV_N_80_PR_1_2_w_tm	osu9725	PAS1136		parallel	Hold	Owens		
> 3130444.owe...	RASPA_convert	osu1842	PA40026	140:50:24	serial	Running	Owens		
> 3130446.owe...	RASPA_convert	osu1842	PA40026	138:30:25	serial	Running	Owens		
> 3130447.owe...	RASPA_convert	osu1842	PA40026	138:09:22	serial	Running	Owens		
> 3133547.owe...	high_yp_PIV_N_80_choke_wo_tm	osu9725	PAS1136	17:36:02	parallel	Running	Owens		
> 3137260.owe...	Case42	osu8290	PA40008	96:36:34	longserial	Running	Owens		
> 3137285.owe...	Case195	osu8290	PA40008	163:01:58	longserial	Running	Owens		
> 3137292.owe...	Case261	osu8290	PA40008	165:44:57	longserial	Running	Owens		

And many more, such as in-browser terminal, job apps, noVNC desktops and apps

Example Current Engagements and Deployments

Production Deployments



In Process of Installing



Come see
my poster tonight
for more info!

Get Started!

- SGCI Affiliate; Listed in SGCI Catlog
- Documentation and code repository available at:
<http://openondemand.org/>
- Send email to ood-users-request@lists.osc.edu
with the subject "subscribe" to join the mailing list
- Webinars and conference publications available
on the website

Open OnDemand website
QR code



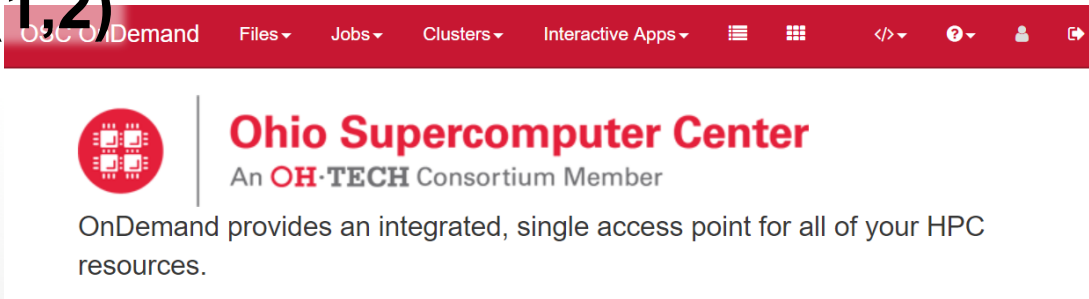
Based upon work supported by the National
Science Foundation under grant numbers 1534949
and 1835725.

OSC OnDemand Overview

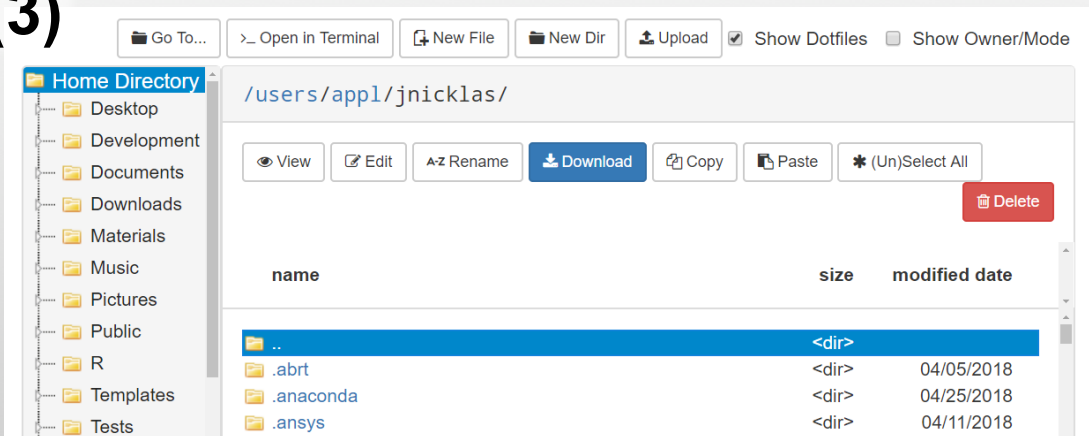
1. Browse to OSC OnDemand (Dashboard)
2. Show navigational elements
3. Launch File Explorer in home directory

4. Launch Active Jobs to show jobs on cluster
5. Launch Shell App
6. Go to Interactive Sessions for Jupyter demo

(1,2)



(3)



(4) Active Jobs

Show entries

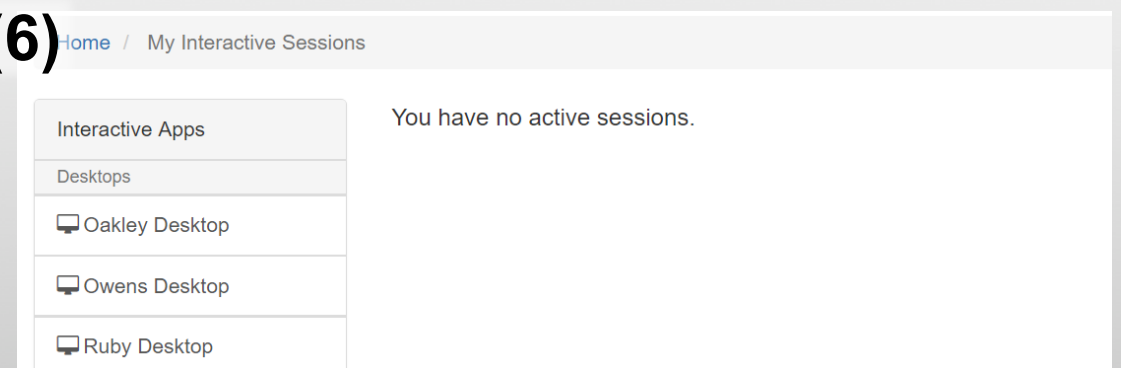
Filter:

ID	Name	User	Account	Time Used	Queue	Status	Cluster
> 317770...	Case333	osu8290	PAA0008	320:23:...	longserial	Running	Owens
> 317771...	Case46	osu8290	PAA0008	324:07:...	longserial	Running	Owens

(5)

```
AS of 2018-05-10T04:02:01.000000 userid jnicklas on /fs/project/PZS0002 used 0 GiB
0 GiB and 11 files of quota 0 files
As of 2018-05-10T04:02:53.000000 userid jnicklas on /users/appl used 216.86 GiB of
GiB and 1144721 files of quota 2000000 files
[jnicklas@owens-login04][~]
```

(6)



Jupyter through OnDemand (1 node | 28 cores)

1. Launch Jupyter from OnDemand
2. Connect to Jupyter when it starts
3. Open a terminal in Jupyter with `htop`
4. Open a Notebook using one of the cluster-installed Anaconda modules

5. Benchmark a Pi calculation while observing resource utilization in realtime (uses single core)

(1) Jupyter Notebook

This app will launch a Jupyter Notebook server using Python on the Owens cluster.

Project

PZS0002

You can leave this blank if **not** in multiple projects.

Number of hours

1

Node type

any

- **any** - (1-28 cores) Use any available Owens node. This reduces the wait time as there are no node requirements.
- **gpu** - (1-28 cores) Use an Owens node that has an **NVIDIA Tesla P100 GPU** and loads the **CUDA 8.0.44** module. There are 160 of these nodes on Owens.
- **hugemem** - (48 cores) Use an Owens node that has 1.5TB of available RAM as well as 48 cores. There are 16 of these nodes on Owens.
- **debug** - (1-28 cores) For short sessions (= 1 hour) the debug queue will have the shortest wait time. This is only accessible during 8AM - 6PM, Monday - Friday. There are 6 of these nodes on Owens.

Number of cores

28

Number of cores on node type (4 GB per core unless requesting whole node). Leave blank if requesting full node.

☐ I would like to receive an email when the session starts

Launch

* All Jupyter Notebook session data is generated and stored under the user's home directory in the corresponding data root directory.

(2) Jupyter Notebook (3230391.owens-batch.ten.osc.edu) 1 node | 28 cores | Running

Host: o0522.ten.osc.edu **Delete**

Created at: 2018-05-09 10:54:54 EDT

Time Remaining: about 1 hour

Session ID: 262c897a-e691-4a17-86ad-cba7c048fe84

Connect to Jupyter

(4)

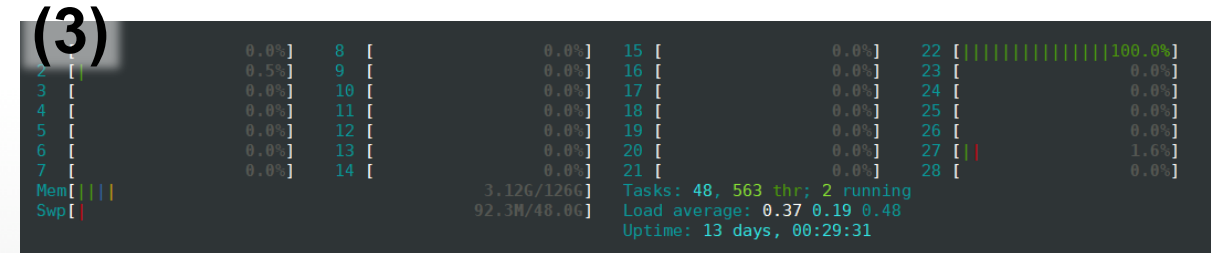
Upload New ↻

Notebook:

- Julia 0.5.1 [julia/0.5.1]
- Python 2.7 [python/2.7]
- Python 3.5 [python/3.5]
- Python 3.6 [python/3.6]
- python36_env [~/conda/envs/python36_env/]

Other:

- Text File
- Folder
- Terminal



(5)

```
import random
def inside(p):
    x, y = random.random(), random.random()
    return x*x + y*y < 1

from functools import reduce
def count(n):
    return reduce(lambda sum, x: sum + 1 if inside(x) else sum, range(0, n), 0)

In [2]: NUM_SAMPLES = 1_000_000_000

In [3]: %time total_count = count(NUM_SAMPLES)
print("Pi is roughly %f" % (4.0 * total_count / NUM_SAMPLES))

CPU times: user 7min 41s, sys: 0 ns, total: 7min 41s
Wall time: 9min 8s
Pi is roughly 3.141680
```

Scaling Science — Jupyter / Spark (4 nodes | 112 cores)

1. Launch Jupyter / Spark from OnDemand
2. Connect to Jupyter when it starts
3. Open a `pyspark` notebook
4. Show number of cores Spark is using
5. Launch same Pi calculation using Spark

(1) Jupyter + Spark

The app will launch a Jupyter Notebook server using Python as well as an Apache Spark cluster on the Owens cluster.

Project

PZS0002

You can leave this blank if not in multiple projects.

Number of hours

3

Number of nodes

4

Node type

any

- **any** - (28 cores) Use any available Owens node. This reduces the wait time as there are no node requirements.
- **hugemem** - (48 cores) Use an Owens node that has 1.5TB of available RAM as well as 48 cores. There are 16 of these nodes on Owens.

Number of workers per node

1

This describes how the cores and memory are divided up on the node (useful to reduce memory allocated for each worker). Should be a multiple of the number of cores on the node you chose above. Do NOT exceed the number of cores on the node.

☐ Only launch the driver on the master node.

This is typically used for `.collect` and `.take` operations that require a large amount of memory allocated (> 2GB) for the driver process.

☐ Include access to OSC tutorial/workshop notebooks.

☐ I would like to receive an email when the session starts

Launch

(4)

```
In [1]: print("Number of cores: %d" % sc.defaultParallelism)
```

Number of cores: 112

(5)

```
In [2]: import random
def inside(p):
    x, y = random.random(), random.random()
    return x*x + y*y < 1

def count(n):
    return sc.parallelize(range(0, n)).filter(inside).count()
```

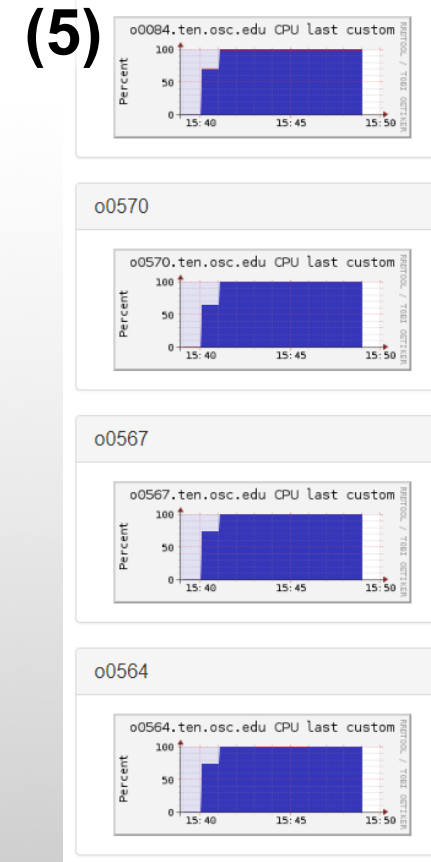
```
In [3]: NUM_SAMPLES = 1_000_000_000
```

```
In [4]: %time total_count = count(NUM_SAMPLES)
print("Pi is roughly %f" % (4.0 * total_count / NUM_SAMPLES))
```

CPU times: user 19.2 ms, sys: 8.49 ms, total: 27.7 ms

Wall time: 6.31 s

Pi is roughly 3.141834



87x speedup

Summary

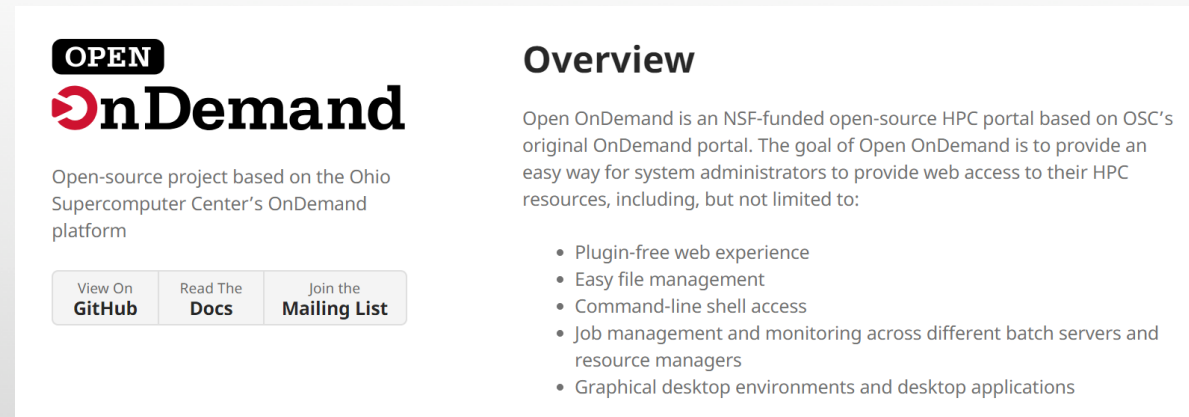
- Jupyter / Spark is a powerful distributed environment
- OnDemand makes it easy to use on an HPC cluster
- JupyterHub + BatchSpawner is a nice Jupyter-only alternative
 - Both rely on reverse-proxy based solution
- OnDemand also makes it easy to use COMSOL Server, RStudio Server, and X11 applications running in a VNC/websockify stack

Visit our website to get started:

<http://openondemand.org>

Join our mailing list to keep in touch:

<https://lists.osu.edu/mailman/listinfo/ood-users>



The screenshot shows the Open OnDemand website. On the left, there is a logo with the word "OPEN" in a black box above a red circular icon with a white arrow, followed by the text "nDemand". Below the logo, it says "Open-source project based on the Ohio Supercomputer Center's OnDemand platform". At the bottom of this section are three buttons: "View On GitHub", "Read The Docs", and "Join the Mailing List". On the right, under the heading "Overview", there is a paragraph describing the project as an NSF-funded open-source HPC portal. Below this, a bulleted list highlights features: "Plugin-free web experience", "Easy file management", "Command-line shell access", "Job management and monitoring across different batch servers and resource managers", and "Graphical desktop environments and desktop applications".

OPEN
nDemand

Open-source project based on the Ohio Supercomputer Center's OnDemand platform

[View On GitHub](#) [Read The Docs](#) [Join the Mailing List](#)

Overview

Open OnDemand is an NSF-funded open-source HPC portal based on OSC's original OnDemand portal. The goal of Open OnDemand is to provide an easy way for system administrators to provide web access to their HPC resources, including, but not limited to:

- Plugin-free web experience
- Easy file management
- Command-line shell access
- Job management and monitoring across different batch servers and resource managers
- Graphical desktop environments and desktop applications