Deploying a Unidata JupyterHub on the NSF Jetstream Cloud, Lessons Learned and Challenges Going Forward

ESIP Summer 2019 Meeting

Julien Chastang

Wednesday, 17 July 2019

Outline

Background and Context

Deploying a Geoscience JupyterHub on NSF Jetstream Cloud

Lessons Learned, Challenges Going Forward

NSF Award # NSF-1344155

Background and Context

Unidata 2024 Proposal: Science as a Service

The Science as a Service concept draws together Unidata's ongoing work to provide geoscience data and software for analysis and visualization with access to workflows designed to take advantage of cloud computing resources.

Award # NSF-1344155

NSF Jetstream Cloud Collaboration

- What is Jetstream?
  - A National Science and Engineering Cloud funded by an $11 million NSF grant.
  - Data centers at IU and TACC.
- Attached to fast Internet2 capability.
- Cloud based on OpenStack for creation of VMs, routers, networks, subnets, security groups etc.
- Unidata has been operating on Jetstream for 3 years through a large research grants
- Once you get through granting process, Jetstream is free

Unidata's Exploration of the Jetstream Cloud Thus Far

- Started by containerizing Unidata technology offerings
  - THREDDS
  - LDM
  - McIDAS ADDE
  - RAMADDA
- Deployed containers to create near complete Unidata data center
- Plenty of NCEP at thredds-jetstream.unidata.ucar.edu
- But what about client-side offerings in cloud?
- *Next obvious step: "data-proximate" analysis and visualization*

Award # NSF-1344155

Deploying a Geoscience JupyterHub on NSF Jetstream Cloud

Award # NSF-1344155

What is a Jupyter Notebook?

A narrative of:

- Explanatory and expository text
- Software code (Python, R, etc.) and output
- Equations (MathJax, LaTeX)
- Figures and multimedia

Lorenz System

The Lorenz system is a series of Ordinary Differential equation studied by Edward Lorenz.

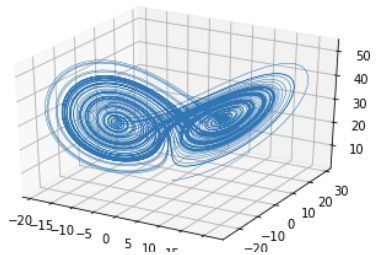
$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

In [10]:
```
def lorenz(x, y, z, s=10, r=28, b=2.667):
    x_dot = s*(y - x)
    y_dot = r*x - y - x*z
    z_dot = x*y - b*z
    return x_dot, y_dot, z_dot

dt = 0.01; stepCnt = 10000
xs = np.empty((stepCnt + 1,))
ys = np.empty((stepCnt + 1,))
zs = np.empty((stepCnt + 1,))
xs[0], ys[0], zs[0] = (0., 1., 1.05)

for i in range(stepCnt):
    x_dot, y_dot, z_dot = lorenz(xs[i], ys[i], zs[i])
    xs[i + 1] = xs[i] + (x_dot * dt)
    ys[i + 1] = ys[i] + (y_dot * dt)
    zs[i + 1] = zs[i] + (z_dot * dt)

fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot(xs, ys, zs, lw=0.5)
plt.show()
```



Award # NSF-1344155

Success of Jupyter in Research and Education

nature
International journal of science

TOOLBOX · 30 OCTOBER 2018

Why Jupyter is data scientists' computational notebook of choice

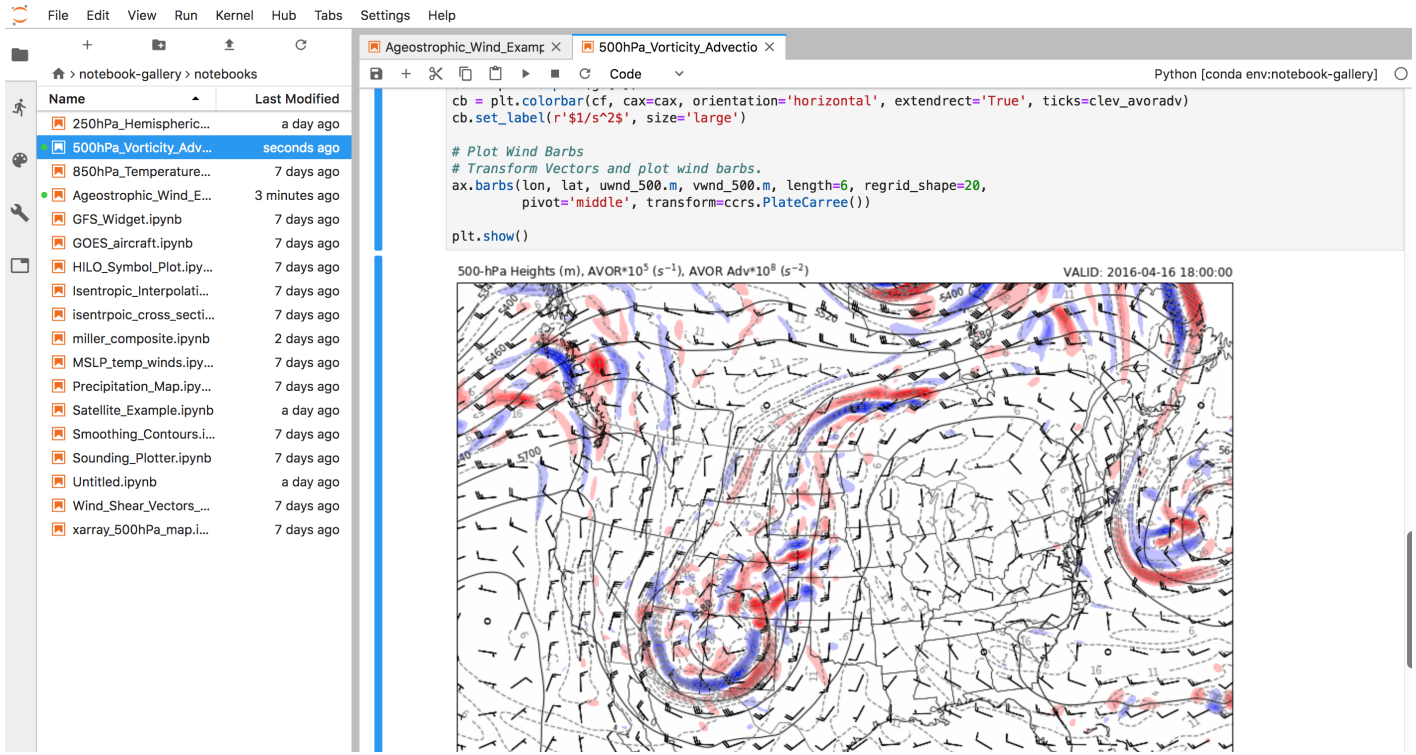
An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.

Jeffrey M. Perkel

[Jupyter] notebooks are really a killer app for teaching computing in science and engineering - Lorena Barba, Engineering Professor, GWU

Award # NSF-1344155

JupyterLab: Next Generation UI

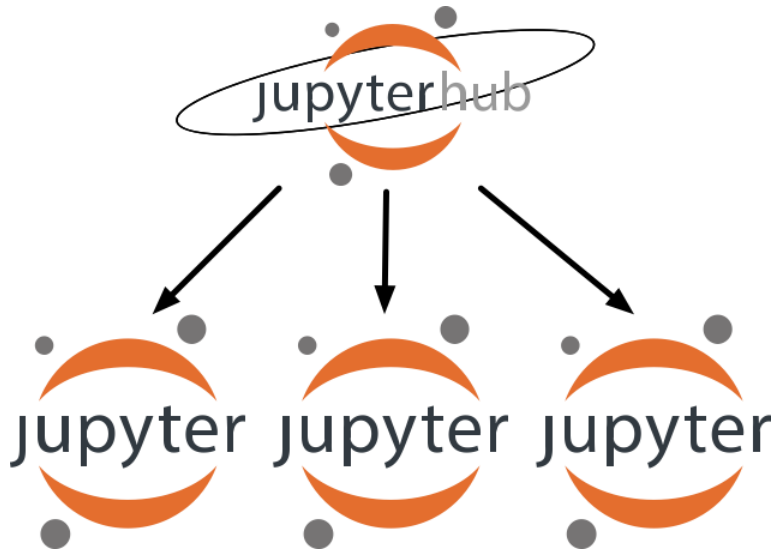


- Terminal (`git`, `conda`, etc.)
- Text Editor

JupyterHub: Multi-user Jupyter Notebook Server

Fernando Pérez: It is infeasible for IT support to assist 800 students install complex software on their laptops.

- Users log in to a JupyterHub server
- Users have their own work space w/ notebooks
- Excellent for workshops or in the classroom
- Administrator can configure ahead of time on behalf of user



Award # NSF-1344155

Zero to JupyterHub Project

Problem: A single JupyterHub server running on a large VM can only serve a small number of students (< 10).

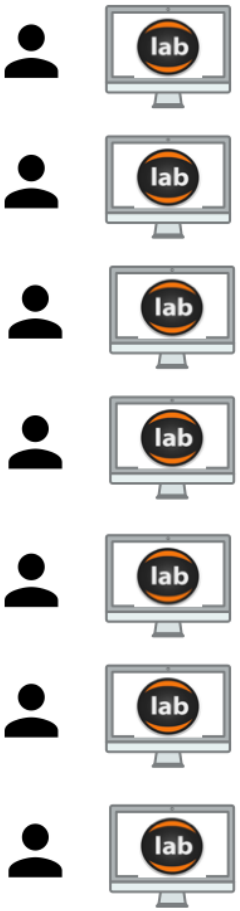
Solution: Zero to JupyterHub project aims to install JupyterHub across several orchestrated VMs to accommodate many more users

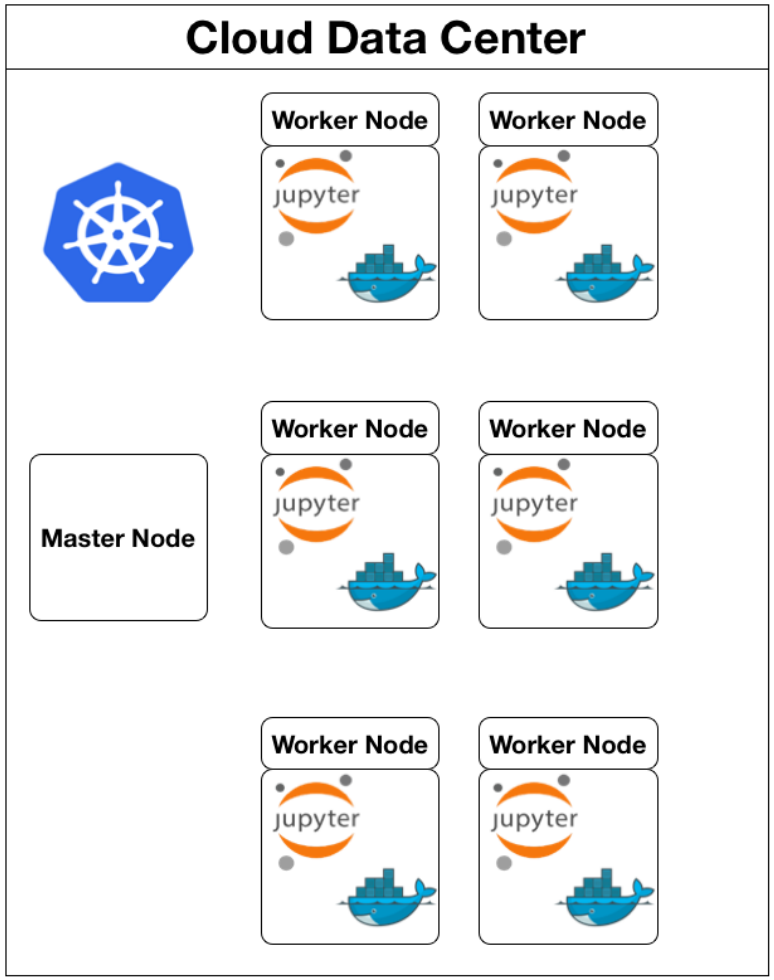
- Virtual Machines
- Software Containers (i.e., Docker)
- Data center software orchestration (i.e., Kubernetes)

Zero to JupyterHub allows for many more users.

Award # NSF-1344155

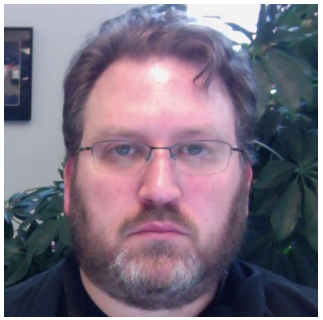
Zero to JupyterHub





Zero to JupyterHub on Jetstream/OpenStack

- z2j ported to Jetstream by Andrea Zonca SDSC, w/ help from Jeremy Fischer at IU



Award # NSF-1344155

K8s Deployed on Jetstream with KubeSpray Project

- Deploy Kubernetes clusters with:
  - Terraform: creation on VMs, routers, networks, subnets, security groups
  - Ansible: kubernetes cluster software installation
- Added layer of scripts to streamline deployment
  - `setup-kube.sh`
  - `setup-kube2.sh`
- Initially, must decide on size/number of VMs via `terraform.tf`
- Can be scaled "manually" thereafter (no autoscaling)

Zero to JupyterHub Customization and Configuration

- YAML configuration file
- HTTPS available with LetsEncrypt or custom certificates
- Authentication via oauth (github, globus, etc.)
- Custom Unidata Docker Container:
  - Gallery: PyAOS examples
  - Workshop: PyAOS training
  - Online Python Training
- Environments to run these projects already installed
- JupyterLab

Award # NSF-1344155

Persistent Storage Allocation for Each User

- Each user gets a 10 GB disk allocation
- This disk space remains available to them for ? amount of time

Award # NSF-1344155

Unidata Community JupyterHub

- jupyterhub.unidata.ucar.edu
- 5 "m1.medium" size VMs totaling 30 CPUs 80 RAM
- 60 users (not concurrent) most of which try it one time, though some return customers

Award # NSF-1344155

Other Unidata JupyterHubs

- Notre Dame of Maryland University (no kubernetes)
- Southern Arkansas University (no kubernetes)
- JupyterHub for UCAR SOARS summer internship program

Award # NSF-1344155

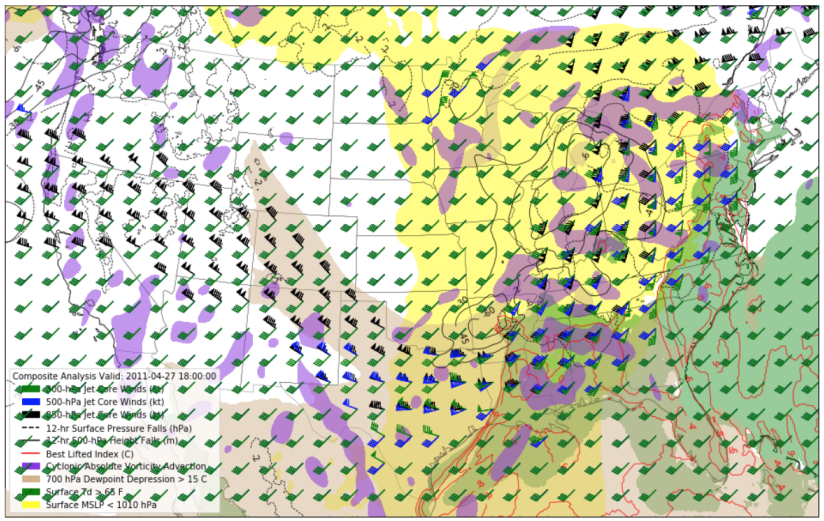
Example Notebook: Miller Composite

Satellite_Exam | 500hPa_Vortic | miller_composi

Code | Python [conda env:notebook-gallery]

```
Legend
rple = mpatches.Patch(color='BlueViolet', label='Cyclonic Absolute Vorticity
llow = mpatches.Patch(color='yellow', label='Surface MSLP < 1010 hPa')
een = mpatches.Patch(color='green', label='Surface Td > 65 F')
n = mpatches.Patch(color='tan', label='700 hPa Dewpoint Depression > 15 C')
d_line = lines.Line2D([], [], color='red', label='Best Lifted Index (C)')
shed_black_line = lines.Line2D([], [], linestyle='dashed', color='k',
                               label='12-hr Surface Pressure Falls (hPa)')
ack_line = lines.Line2D([], [], linestyle='solid', color='k',
                        label='12-hr 500-hPa Height Falls (m)')
t.legend(handles=[jet300, jet500, jet850, dashed_black_line, black_line, red
                  purple, tan, green, yellow], loc=3,
         title='Composite Analysis Valid: {:s}'.format(str(vtimes[0])))

t.show()
```



Example Notebook: Satellite + GFS Model

Satellite_Exam | 500hPa_Vortic | miller_composi

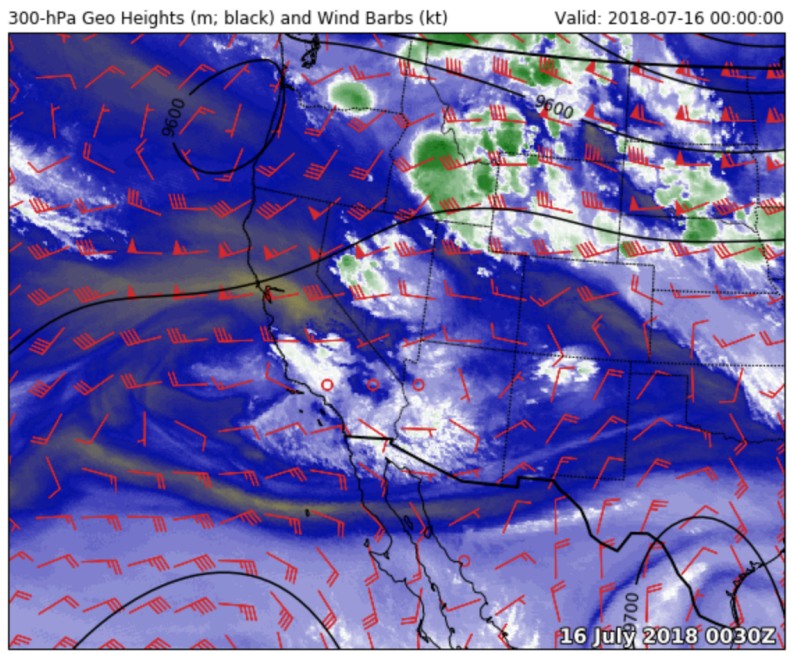
Code | Python [conda env:notebook-gallery]

```
ax.set_extent([-132, -95, 25, 47], ccrs.Geodetic())
cs = ax.contour(lon, lat, Z_300, colors='black', transform=ccrs.PlateCarree(
ax.clabel(cs, fontsize=12, colors='k', inline=1, inline_spacing=8,
          fmt='%i', rightside_up=True, use_clabeltext=True)

ax.barbs(lon, lat, uwnd_300.to('knots').m, vwnd_300.to('knots').m, color='ta
         length=7, regrid_shape=15, pivot='middle', transform=ccrs.PlateCar

ax.set_title('300-hPa Geo Heights (m; black) and Wind Barbs (kt)', loc='left
ax.set_title('Valid: {}'.format(vtime[0]), loc='right')

plt.show()
```

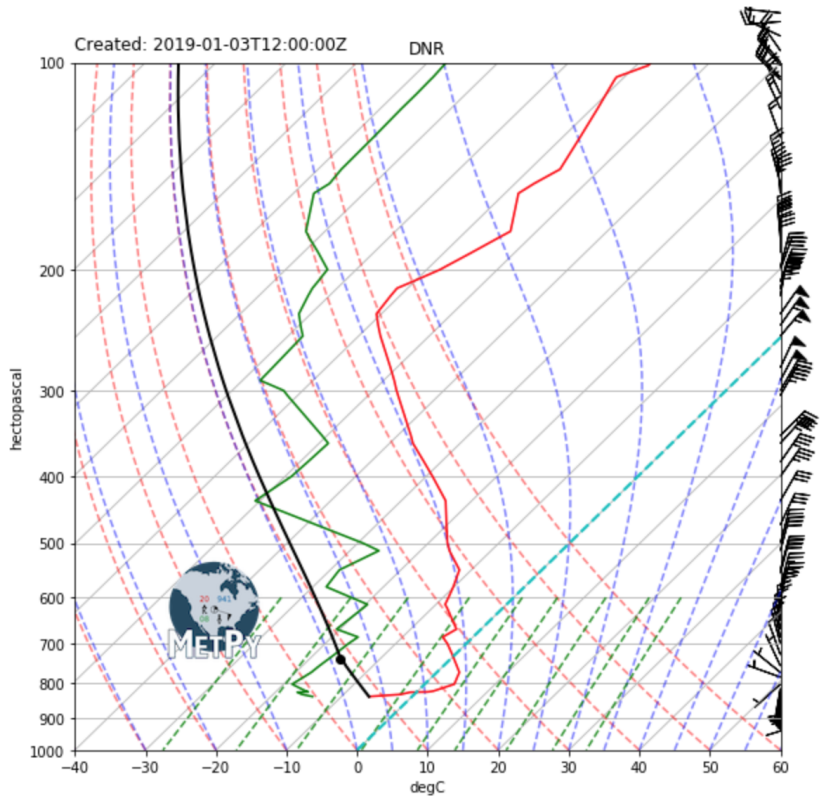


Example Notebook: Upper Air SkewT

Sounding_Plotter.ipynb

Code | Python [conda env:notebook-gallery]

```
        with tempfile.NamedTemporaryFile(suffix='.png') as f:
            skewt.ax.figure.savefig(f.name)
            uploader.upload_to(f.name, posixpath.join(args.gdrive, fname))
    else:
        skewt.ax.figure.savefig(make_name(args.site, request_time))
```



Lessons Learned, Challenges Going Forward

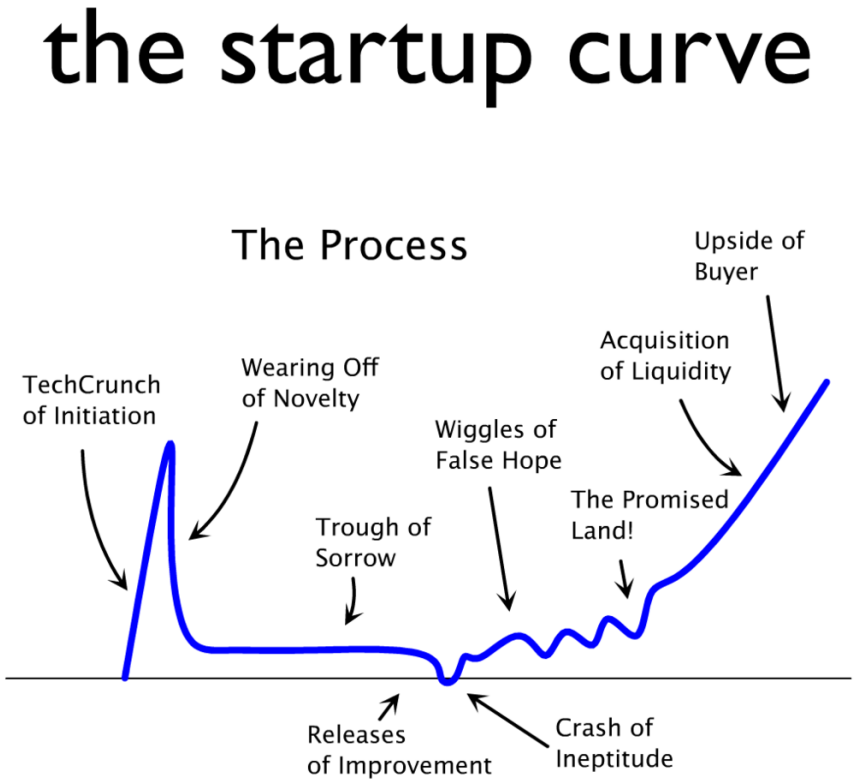
Award # NSF-1344155

Problems with JupyterHub Deployments and Maintenance

- Finicky VMs during cluster creation (takes several tries)
- Timeout errors at every level (deployment and running)
- Complexity associated with software running on clusters
- JupyterHub spawn errors
- Disk allocation errors
- Network problems at TACC
- General lack of reliability throughout entire tech stack

Award # NSF-1344155

Trough of Sorrow



Source: Paul Graham via andrewchen.co

Additional Caveats

- What to do with user data over long term? What guarantees?
- This project is not Pangeo
  - Goals are more modest and Unidata focused
  - Would be happy to deploy Pangeo on Unidata Jetstream allocation

Award # NSF-1344155

Lessons Learned

- Scriptifying your deployments to make your life easier
- Interview your users before so cluster can be accurately sized
- Tech stack is new and fragile and will take time before it is stable
- Don't advertise too early, scale gradually by introducing to incrementally wider audiences to address problems
- Need to be persistent to overcome tech challenges: ask for help on github issues and gitter
- Science professionals have a high threshold for problems as long as they can arrive at a desired objective

Award # NSF-1344155

Future Plans

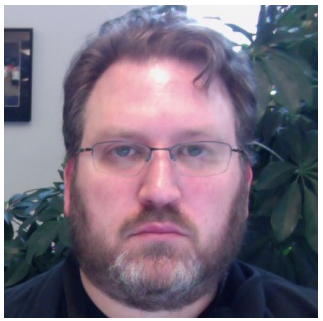
- Get technology problems under control, don't build on a Swiss cheese foundation
- Experiment more with IU Jetstream data center
- Autoscaling on OpenStack with Zonca collaboration
- Address github issues
- Once things stabilize, promote to wider community

Award # NSF-1344155

Acknowledgments

We thank Brian Beck, Maytal Dahan, Jeremy Fischer, Victor Hazlewood, Peg Lindenlaub, Suresh Marru, Lance Moxley, Marlon Pierce, Semir Sarajlic, Craig Alan Stewart, George Wm Turner, Nancy Wilkins-Diehr, Nicole Wolter and Andrea Zonca for their assistance with this effort, which was made possible through the XSEDE Extended Collaborative Support Service (ECSS) program.

Special thanks to Andrea And Jeremy



Resource and Questions

https://github.com/Unidata/xsede-jetstream

Award # NSF-1344155