Data Science in US and Canadian Higher Education

OR

Enabling Educational Infrastructure for Jupyter

Laura Norén
Director of Research, Obsidian Security
laura.noren@nyu.edu
@digitalFlaneuse

Anthony Suen
Director of Programs,
UC-Berkeley Division of Data Sciences
anthonysuen@berkeley.edu
@Anthony_Suen

Berkeley
UNIVERSITY OF CALIFORNIA

23 August 2018 | New York, NY

jupytercon

Thank you

To the Moore-Sloan Data Science Environment

To my co-author Anthony Suen of UC-Berkeley

To the Jupyter team

Our challenge.

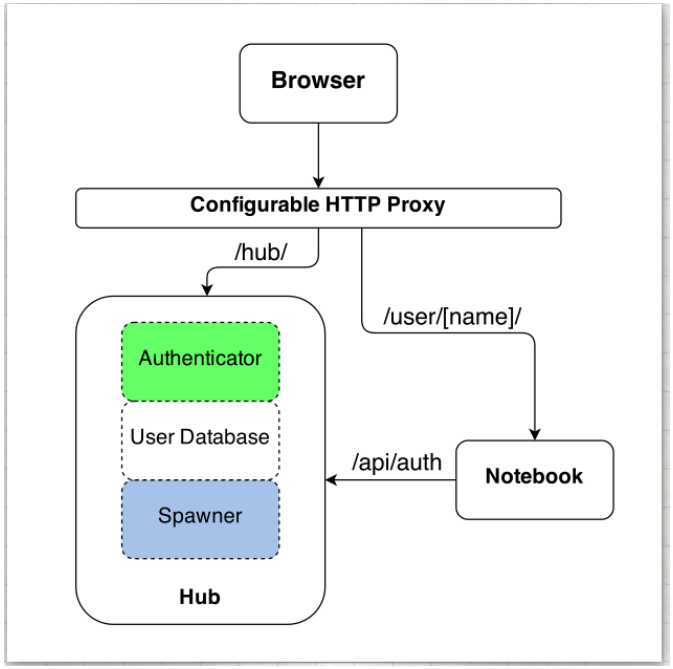
How can we teach all the students who want to learn computer science, statistics, and data science in a way that is:

a. pedagogically compelling
b. good runway for the work students will do
c. reproducible
d. scalable
e. affordable
f. doesn't burn out instructors

What is a JupyterHub?

Three subsystems make up JupyterHub:

- a multi-user Hub
- a configurable http proxy
- multiple single-user Jupyter notebook servers (Python/IPython/tornado)



What does a JupyterHub do?

JupyterHub performs the following functions:

- The Hub launches a proxy
- The proxy forwards all requests to the Hub by default
- The Hub handles user login/authentication and spawns single-user servers on demand
- The Hub configures the proxy to forward URL prefixes to the single-user notebook servers

TL;DR A JupyterHub is extremely useful for teaching because it provides a unified environment.

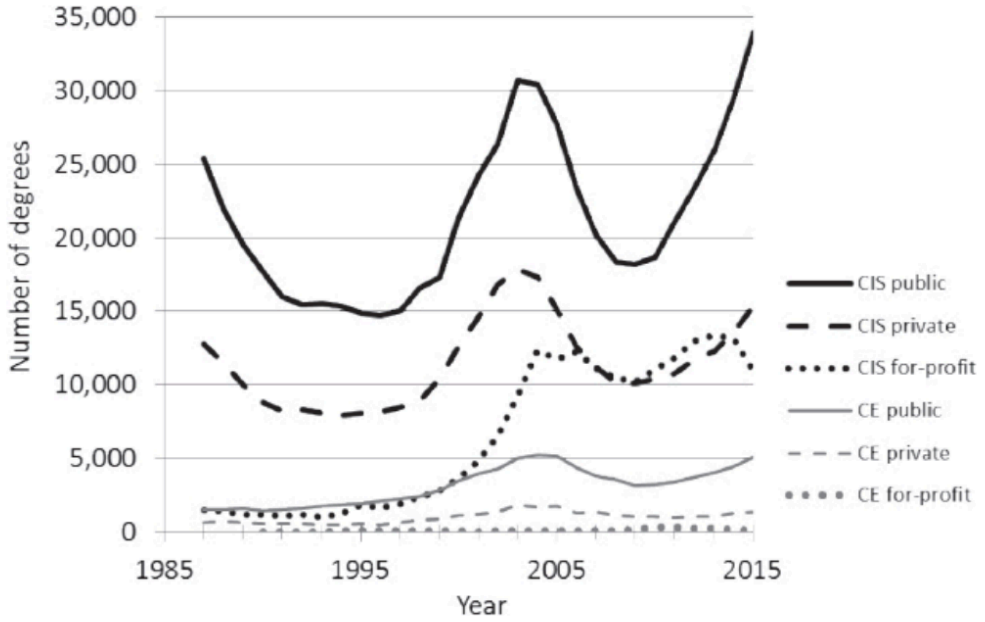
Why the need for hubs?

Universities are experiencing huge demand for undergraduate (and graduate) coursework in:

- **computer science**
- **statistics**
- **data science**

Universities are not experiencing a huge increase in funding.

Bachelor's degree production from 1987 to 2015 in CIS and CE at public, private, and for-profit institutions reporting to IPEDS



Why the need for hubs?

Using JupyterHubs allows existing courses to scale while maintaining rigor and quality.

UC-Berkeley's Data8 class - foundations of data science - had over 1000 students enrolled last fall.

Methods: Qualitative interviews

Interviews with 12 representatives who have worked to establish Jupyter Hubs at their universities.

Schools ranged in size from small liberal arts to large public universities.

Case 1: Berkeley

Who is on their installation and support team?

2 tenured professors, 1 full-time staff member, ~10 postdocs and grad students who can help troubleshoot

How big are the classes?

1,000 students in data8, plus 10,000+ in a free online EdX version

How are they handling installations?

Team is competent, able to work closely with IT

How do they pay for it?

Cloud credits from two of the big three (Microsoft, Google)

Case 1: Berkeley - the challenges

Team problems:

Careers:

Graduate students and postdocs are temporary, have to have other priorities to advance their careers.

There isn't enough funding to pay them well enough to live in the Bay Area.

Sustainability, scalability problems:

Relying on free cloud credits could be precarious.

It's hard to share free cloud credits with other schools.

Case 2: Small liberal arts university

Who is on their installation and support team?

Usually, a professor or two

How big are the classes?

~20-30 students

How are they handling installations?

Big struggle…departmental resources, ask Berkeley??

How do they pay for it?

Tough, ad hoc. Not a lot of funding to support computing for 'typical' teaching.

When cloud providers give credits directly to students, it doesn't scale.

Case 2: Small liberal arts university - outcome?

Team problems:

Careers:

Professors are overburdened and it's difficult for them to find time to work with IT departments.

Sustainability, scalability problems:

Lots of professors may give up on JupyterHubs altogether.

Case 3: Wealthy private university

Who is on their installation & support team?

IT professional surrounded by other IT professionals

How big are the classes?

~12-50

How are they handling installations?

"we hired a firm to help us implement Jupyterhub in Amazon AWS cloud"

Case 3: Wealthy private university

How do they pay for it?

The university covers all costs from general funds.

They moved from using a Docker instance per student to using an EC2 instance, bringing costs for small classes from $15 per student to $3 per student

> "With EC2 its min $34 - max $717/month for 20 users."

They have a GitHub Repo with all of their installation code, including for the EC2 option.

See also: GitHub Repo explaining how to calculate costs.

Case 3: Wealthy university - troubles?

Replicability, scalability:

Many schools cannot rely on their university's operating budget to support this type of teaching expense.

Their classes were still relatively small (12-50 students). When they scale, costs will grow. Even though they are a wealthy school, there is pressure to keep costs low.

Case 4: Canadian Federation (PIMS)

Who is on their installation and support team?

1 full-time System Network Manager,

time donation from profs. at 10 different institutions

How big are the classes?

200-300 students

How are they handling installations?

System Network Manager works with Compute Canada

How do they pay for it?

"The program and activity was really bootstrapped based on staff time"

The System Mgr. is paid for by Compute Canada. Grants from Canadian federal govt ($4.5m) and Alberta ($1m) keep profs, teachers supported, sort of.

Large goodwill/volunteer halo around paid positions.

Case 4: Canadian Federation (PIMS) - troubles

Careers:

- Still relies on a lot of donated faculty time BUT they seem to be rewarded for this work.

Scaling:

- They may not be able to handle >1000 students at a time

Still a highly functional, potentially replicable model

- Can accommodate small classes, high schoolers
- Funding is a hurdle, not a wall
- Teachers can focus on course development
- Ideas, course modules shared in network

We support a multi-level approach

Littlest Jupyter Hubs for small schools - Yuvi Panda

- No need to set up Kubernetes - eliminates a pain point
- Good for <50 students - this describes many classes
- Could potentially run on a local server (rarely done, but may avoids the cloud credit need)

Federated distribution model on a regional basis

Federated distribution model - inspired by Canada

Why a federation?

Can more efficiently establish the infrastructure to support large classes (e.g. Kubernetes)

Partially centralizes collection and distribution of cloud credits

Partially centralizes collection and distribution of best practices in teaching

- Journal of Open Source Education already helps publish educational material

Federated distribution model - inspired by Canada

What's the federation?

National Science Foundation Big Data Innovation Hubs and Spokes (4 regions)?

Pros: Theoretically, this covers every school. Takes advantage of existing network.

Easier for large cloud computing companies to donate credits - only 4 key players.

Cons: The NSF Big Data Hubs don't currently have the staff support for this.

State university systems (50 exist, some are more capable than others)?

Pros: States may be better able to collect state-based grants.

Cons: May leave out weaker states and private colleges.

Big picture: Why set up JupyterHubs at all?

National imperative

Educating the STEM workforce is a national imperative.

Steady IT and cloud credit support can scale to small institutions

Small liberal arts schools that give up can either link to regional or national hubs or set up Littlest Jupyter Hubs. Calling Berkeley isn't sustainable.

Postdoc/grad student labor is misaligned

Relying on postdocs and grad students is precarious project management

Postdocs and grad students typically do not advance their careers by doing SysAdmin work.

Thank you.

Laura Norén, laura.noren@nyu.edu
Anthony Suen, anthonysuen@berkeley.edu