Jupyter and HPC: Current State and Future Roadmap

IDEAS-ECP Best Practices for HPC Software Developers Webinar Series
Date: February 28, 2018
Presented by: Matthias Bussonnier, Suhas Somnath, and Shereyas Cholia

---

Questions and Answers from the Webinar

Q. I heard that JupyterLab is gonna release the 1.0 version and it seems that JupyterLab and Jupyter are gonna converge to one?

A: As noted in the presentation Jupyter is a protocol and a set of standard, there are many software that implement the protocol, so the only thing we can do a recommand an official version. The current notebook interface, we refer to as the "Classic notebook" will be retiring in a few years. We will maintain both JupyterLab and Classic notebook for the time being – and you can use both at the same time. JupyterLab ultimately will have new features that won't be available in the classic notebooks – at leat the core team will decrease involvment. As the software is open source this will of course depends on volonteers and complanies involvment.

Q. How will JupyterLab and CoCalc evolve together?

A: CoCalc is a for-profit company, so it will be up to them. The Jupyter team has occasional conversations with them, but it is not the same project. The driving force behind both can be quite different, CoCalc can have a faster implementation/deployment cycle depending on the customer use as they control the full deployment. JupyterLab being open-source and used by many other products, needs to have a more careful development cycle.

Q. In Jupyterlab, is there an option to have matplotlib plots pop-up in a separate window within the Jupyterlab interface (instead of being inlined in a notebook) ?

A: Part of future development. You should be able to right-click in an output to open a linked-copy in a new panel. Pull requests are welcome on github.com/jupyterlab/jupyterlab.

Q: Have you considered to include some of the features from Spyder into JupyterLab, such as the variable view window, which resembles Matlab and I think it's very convenient?

A: Part of future development. One of the challenge is to make that multi-language aware. Making it for a single language can make assumption that Jupyter cannot. You can find ad-hoc extension (workspace view) for some languages. What we ship is a curated collection of extensions but a number of other community maintained and often hi-quality extension also exists.

Q: I'm an author/maintainer of a (Coarray) Fortran Kernel. (I don't really know what I'm doing and would welcome any tips or suggestions.) Are there API changes that will require kernel maintainers to update their kernels that currently run in Jupyter Notebooks to allow them to run in Jupyter Lab?

A: No you don't need to change anything. Also don't assume the Core Jupyter team started really knowing what we were doing. Not sure we do now. If you have any feedback on the API, you are welcome to ask us.

Q: You mentioned in the slides that Jupyter could be run interactively on a cluster. Could you send a link to the corresponding documentation?

A: There are a number of moving part with corresponding documentation, you will need to understand 1st Jupyterhub (jupyterhub.readthedocs.org), that what allow multiple users to each run securely and easily notebook servers. Authenticator are pluggable pieces of JupyterHub that hook into Authentication and Spawner are pluggable pieces that configure how you can start servers for users.
Once JupyterHub is in place you need to understand the distinction between the notebook server and the kernels, one common misconception is that 1 kernel == 1 language, though 1 kernel is a set of configuration , it can be a language, but it can be :

- Hardware requirement
- Location (eg a specific beamline , like brookhaven does)
- A specific queue…
- Etc.

You thus can (for example), set up multiple kernels that start the same environement with different configuration (you'll do that via kernelspecs which are json files (http://jupyter-client.readthedocs.io/en/stable/api/kernelspec.html)), for example using remote_ikernel (https://pypi.python.org/pypi/remote_ikernel), the readme has lot of informations. Your user(s) will now have access in the menu into various way of starting a different kernel in various ways.

Bonus comment: Of possible interest regarding CERN's Swan service for Jupyter analysis in their cloud: https://swan.web.cern.ch/
At the recent EOS workshop at CERN there was a presentation which included a docker stack running eos + cernbox + Swan available to explore the services on your own:
https://cernbox.cern.ch/cernbox/doc/boxed/
The presentation for which is available at:
https://indico.cern.ch/event/656157/contributions/2866305/attachments/1595576/2527036/EBocchi_EOS_Workshop.pdf

Q: Has there been any discussion about DOIs for notebooks?

A: It is in the process of evaluation.

Q: Who is pushing "All papers" to have notebooks and/or Data?

A: At the moment - Institute for Functional Imaging of Materials (at ORNL) though we are trying to get more institutes onboard and expand to the rest of the lab.

Q: Does CADES have JupyterHub deployed already ?

A: One specific group has deployed JupyterHub on the Compute and Data Environment for Science (CADES) on OpenStack. CADES is evaluating a larger lab-wide deployment.

Q: Any cluster teams out there already have Jupyterhub deployed fully at their centers ?
A: Yes! See the third part of the webinar by Shreyas Cholia from NERSC

Q: How does the data get into the notebook? Is that handled by the Python package?

A: It may depends what you mean by "in the notebook". If you `load()` data, it will not get into the notebook document, the way the Jupyter team think about notebooks is that the unit of sharing should be bigger than a notebook (a Git repository ?). It also kinda solve another issue whic is that data lives (often) on a machine which is different than the notebook server and he notebook themselves. Another answer is to use a content adressable data source (dat, quilt, ipfs, …) or somethign like globus, dash, that can assign a DOI to a dataset.

At NERSC the data is on the global file-system visible to the notebook so you can access that directly via python functions for IO.

Q: What is the NERSC deployment doing about internal encryption? Hub <-> Notebook <-> Kernel

A: Currently kernel is on same node as notebook and just uses default zeromq connection to communicate. However, once we split up the kernel and go out over the network between kernel<-> NB encryption will become more important. This is something that will need to be looked at in more detail. Zeromq does support encryption in theory so it may need to be implemented within the Jupyter infrastructure in the future.

Q: Does ORNL/OLCF currently provide access to a JupyterHub (in a similar way as NERSC does)?

A: The OLCF does not currently have a jupyterhub service available, but we have a work-in-progress study ongoing to find a way to run it in a recently deployed openshift (kubernetes) cluster that meets our security policy and HPC network constraints.

Q: Are there plans for NERSC to provide access to Edison's $SCRATCH from a Jupyter notebook ? (Right now, I think that only Cori's $SCRATCH is accessible.)

A: Probably not. Current focus is on CORI. It will depend on demand - we may be able to mount Edison scratch on Cori if needed.

Q: Does it work with two-factor authentication? [Presumably NERSC's deployment]

A: Yes - will work transparently - Just enter OTP + password in login box. No changes needed to Jupyter since this is implemented at the PAM stack level.