Birds of a Feather session on Science platforms

William O'Mullane[1], Megan Sosey[2], Hassan Siddiqui[4], Gregory Dubois-Felsmann[3], Gerard Lemson[5], Christophe Arviset[6], Mike Fitzpatrick[7], Ivelina Momcheva[2], Sebastien Fabbro[8], Brian Major[8]
[1] *Large Synoptic Survey Telescope, Tucson, AZ, USA;* `womullan@lsst.org`

[2] *Space Telescope Science Institute*

[3] *IPAC, California Institute of Technology, Pasadena, CA, U.S.A.*

[4] *Vega for Gaia/ESAC*

[5] *The Johns Hopkins University*

[6] *ESAC Science Data Centre*

[7] *NOAO*

[8] *CADC*

Abstract. How users will interact with data in the future is always unclear. Currently we see Jupyter Notebooks or JupyterLab emerging in many places as the way forward for one aspect of this. This BoF explored some topics around providing and environment for doing science.

1. Introduction

It seems timely to consider how we might offer users a smoother experience as they move between data providers. Current VO services allow one to send queries to multiple centres but in the notebook environment one may wish to do something more sophisticated. We should consider whether users can send requests from one centre to another or whether the same notebooks should be runnable in different centres. How do we deal with batch processing - large jobs? How do we manage resources/quotas (disk/memory/cpu)? How can we enable users to share their work (both notebooks and data) and create ad-hoc scientific collaborations? We had a few short presentations which are all in github[1]:

- LSST Approach (Dubois-Felsmann): science platform will give access to the data and visualization tools and documentation, it will allow collaboration and allow for added value processing close to the data using Jupyter. In principle you could write C++ or any other language in that system.

[1] `https://github.com/lsst-dm/adass27-womullan/tree/master/presentations`

- SciServer Approach (Lemson): SciServer is format agnostic storage with extensible tools (query and analysis), it allows hosting and sharing datasets. Near data access is provided with Jupyter. Notebooks can be executed in batch mode but no MPI.
- ESAC Science Data Centre (Arviset): Science Exploitation and Preservation Platform intends to provide an data computing environment close to the ESAC science archives, enabling scientists to run their code where the data reside, and share their results and code with other people. In addition, the platform will offer an environment to easily run legacy software from ESA space science missions.
- NOAO approach (Fitzpatrick): Data Lab provides full sky exploration of images/catalogs and approximately 1 PB for local user storage (including personal databases), it allows workflows to run close to the data and will soon offer data-publication services. Provides Jupyter notebooks and legacy code execution as containerized applications run in sync/async mode. User support includes docs, example notebooks, web forum and helpdesk contact; few users so these haven't been heavily exercised yet. Open for new users but still in development.
- STScI DSMO (Momcheva): increase science output from holdings, shorten turn around time, connect multi wavelength, considering a Jupyter hub system deployed on Amazon. Cost would be relatively cheap, and some of that cost may be moved to users.
- CADC (Fabbro): Raw OpenStack portal with vanilla VMs, some projects using Jupyter, intending to containerize.
- CADC/IVOA (Major): IVOA and remote computing grid and web services working group, working with knowledge discovery group to define use cases. Goal is fast interoperable computing services close to the data, and support for Machine Learning.

A few good principles were mentioned by Mike Fitzpatrick:

- multiple entry points into the system (web, notebooks, cmdline tools, scripting apis)
- language-agnostic (python flask micro-services architecture, restful interface)
- enable user developed tools
- established standards with hidden complexity for friendly interfaces
- provide access to external data/services vs local ingest

2. Discussion

Current VO services allow one to send queries to multiple centres but in the notebook environment one may wish to do something more sophisticated. We should consider whether users can send requests from one centre to another or whether the same notebooks should be runnable in different centres. How do we deal with batch processing

- large jobs? How do we manage resources/quotas (disk/memory/cpu)? How can we enable users to share their work (both notebooks and data) and create ad-hoc scientific collaborations?

A show of hands, who is doing a project like the above with Jupyter lab/hub showed roughly 50-50 response of those using JupyterLab as opposed to not using. There were about 70 people in the room and perhaps 30 did not raise their hand for either option.

Security was discussed: SciServer is by account but anyone can register and there is no check on Users. Data Lab is actually checking on people who apply for accounts, the APIs authenticate to services, you can still run an anonymous notebook. The advantage of having an anonymous server is for tutorials and EPO projects.

What about authentication? Many use github authentication. This is a concern is that forcing people to use any one of multiple authentication services. This is not a good idea. Agreed, will probably provide multiple authentication services, hopefully will get to an automatic way of authentication via API for science user validation. You will still need an "account" the real question is how that is created i.e. using your credentials from an existing academic organization. Data rights can be complicated as well.

An NCSA research scientist (Nathan Goldbaum), mentioned there's an effort going on there WHOLETALE [2] which is a docker stack that allows you to deploy on your own hardware. This uses docker swarm as the backend balancing system. It is used as a platform for data sharing and computing on top of hardware the user controls.

There is a change in paradigm for data centres, providing data is different than providing processing time. Do we need to consider proposal for type of compute with i/o and sources, with a committee decides what you get? LSST is indeed thinking about these things for the future. CADC say let's first get users, the community is not enormous, they have rough estimates of their needs and are not furtive, so far these worries haven't been a problem. Resources essentially were not a problem on Skyserver/CasJobs either.

It is important to open doors to reprocess images, going to cloud services gives a new user space to provide resources through other methods. Can the containerization run on multiple infrastructures, with great portability? This seems to be where these things are heading!

How can we combine and connect all the resources in the future (the dream of the IVOA)? How can we communicate between the large centers better? That was the point of this BoF. Maybe we should look at the channels others outside of the astronomy community have used? At least in USA there is the national data services consortium, all sciences, had discussions there about containerization for mutually beneficial interactions, NDS has one or two meetings a year: `http://nationaldataservice.org`

The other interesting topic raised was reusing and bringing old code to the new platforms mentioned by NOAO and ESAC. ESAC and NOAO are places with legacy processing software that needs specific machines to run on. Porting CASA, complicated with GUIs. One way to preserve the software is to make a VM available and use VNC in the browser to access the GUI (not great but it does work). STScI find it important to educate our users on new and updated software instead of having them continue to run legacy code. Also rather than move code to data, provide virtual storage space for use,

[2] `http://wholetale.org/`

you just move your data. Okay for small datasets, but helps you avoid rewriting code just to use the new interface.

3. Conclusion

Continue the conversation on SLACK at `https://scienceplatforms.slack.com/` or join the IVOA GWS list (`mailto:grid@ivoa.net`).