Birds of a Feather session on Science platforms

William O'Mullane[1], Megan Sosey[2], Hassan Siddiqui[3], Gregory Dubois-Felsmann[4], 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, Baltimore, MD, USA*

[3]*Vega for Gaia/ESAC, Madrid, Spain*

[4]*IPAC, California Institute of Technology, Pasadena, CA, USA*

[5]*The Johns Hopkins University, Baltimore, MD, USA*

[6]*ESAC Science Data Centre, Madrid, Spain*

[7]*NOAO, Tucson, AZ, USA*

[8]*CADC, NRC Herzberg, Victoria, BC, Canada*

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 an 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 on GitHub[1]:

LSST Approach (Dubois-Felsmann). LSST (Ivezic et al. 2008) Science Platform (Jurić et al. 2017) will give access to the data and visualization tools and documentation, we see three aspects to this, the traditional Portal, the Web Services APIs and the Jupyter/Notebook. It will allow collaboration and allow for added value processing

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

close to the data using Jupyter. In principle you could write C++ or any other language in that system, it will come with the DM stack (Jurić et al. 2016) pre-loaded.

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 type processing is available at the moment.

ESAC Science Data Centre (Arviset). The Science Exploitation and Preservation Platform intends to provide a 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 (Fitzpatrick et al. 2016) 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). We aim to increase science output from holdings, shorten turn around time, connect multi wavelength resources, and we are considering a Jupyter hub system deployed on Amazon. Cost would be relatively cheap, and some of that cost may be moved to users.

CANFAR (Fabbro). CANFAR has been providing user object-like storage through VOSpace, VM on demand, group management and user VM batch processing since 2010 (Gaudet et al. 2010). It provides user support including documentation, help desk, and a slack channel. An increasing number of teams setup their own JupyterHub and container management on top of VMs. A proposal has been submitted to provide containerized workflows, tiered user storage with file-system access, user database-as-a-service, all integrated to a Jupyter based frontend.

CADC/IVOA (Major). The IVOA remote computing grid and web services working group is working with the knowledge discovery group to define use cases. The goal is fast interoperable computing services close to the data, and support for Machine Learning.

2. Discussion

A few good principles were mentioned by Mike Fitzpatrick:

- multiple entry points into the system (web, notebooks, command line 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.

A show of hands by those doing a project like that described in the introduction above showed roughly a 50-50 split of those using Jupyter as opposed to not using it (with LSST being the only users of JupyterLab). There were about 70 people in the room and perhaps 30 did not raise their hand for either option.

Security was discussed: SciServer access is by account only but anyone can self register and there is no check on registrations. Data Lab is actually checking on people who apply for accounts, the APIs authenticate to services, although 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, we do not wish to force people to use any one of multiple authentication services. All agreed that this is not a good idea. We will probably have to 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. For example, that could be done using your credentials from an existing academic organization. Data rights can be complicated as well – we did not delve in to that topic too far in this forum.

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 proposals for a type of compute with I/O and sources, where a committee then decides which proposals merit system time? LSST is indeed thinking about these things for the future but wishes to see how things evolve first. 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 though they provided limited near data processing only.

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 the 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`

[2] `http://wholetale.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 was complicated with GUIs, GUIs are in general problematic. 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, 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`).

Acknowledgments. Some of this material is based upon work supported in part by the National Science Foundation through Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.

References

Fitzpatrick, M. J., et al. 2016, in Software and Cyberinfrastructure for Astronomy IV, vol. 9913 of Proc. SPIE, 99130L

Gaudet, S., et al. 2010, in Software and Cyberinfrastructure for Astronomy, vol. 7740 of Proc. SPIE, 77401I

Ivezic, Z., et al. 2008, ArXiv e-prints. `arXiv:0805.2366`

Jurić, M., Ciardi, D., & Dubois-Felsmann, G. 2017, LSST Science Platform Vision Document, LSE-319, LSST Docushare. URL `http://ls.st/LSE-319`

Jurić, M., et al. 2016, in ADASS XXV, edited by N. P. F. Lorente, vol. TBD of ASP Conf. Ser., TBD. `arXiv:1512.07914`