Astroinformatics
Proceedings IAU Symposium No. 325, 2016
M. Brescia, S.G. Djorgovski, E. Feigelson,
G. Longo & S. Cavuoti, eds.

doi:10.1017/S1743921317001363

What will the future of cloud-based astronomical data processing look like?

Andrew W. Green, Elizabeth Mannering, Lloyd Harischandra, Minh Vuong, Simon O'Toole, Katrina Sealey and Andrew M. Hopkins

Australian Astronomical Observatory,
PO Box 915, North Ryde, NSW, 1670 Australia
email: andrew.green@aao.gov.au

Abstract. Astronomy is rapidly approaching an impasse: very large datasets require remote or cloud-based parallel processing, yet many astronomers still try to download the data and develop serial code locally. Astronomers understand the need for change, but the hurdles remain high. We are developing a data archive designed from the ground up to simplify and encourage cloud-based parallel processing. While the volume of data we host remains modest by some standards, it is still large enough that download and processing times are measured in days and even weeks. We plan to implement a python based, notebook-like interface that automatically parallelises execution. Our goal is to provide an interface sufficiently familiar and user-friendly that it encourages the astronomer to run their analysis on our system in the cloud—astroinformatics as a service. We describe how our system addresses the approaching impasse in astronomy using the SAMI Galaxy Survey as an example.

Keywords. astronomical data bases: miscellaneous, methods: data analysis

1. Motivation

AWG is leading the SAMI Galaxy Survey's first major data release, the Emission Line Physics Release. The Survey is an integral-field spectroscopic survey of nearby galaxies, and includes spectroscopic data cubes as the major data product. The Release includes data for 772 galaxies—approximately one-quarter of the full sample planned. The total data volume is 1.4 terabytes.

While the Releases 1.4 TB data volume is relatively modest compared to many planned and current surveys, the SAMI Team is struggling with the size of the data set. The data takes approximately 10 hours just to copy from one location to another. Just starting the Team's data reduction software (without doing any processing) takes over an hour. Although much of the problem is embarrassingly parallel, only some steps have been parallelised. Many of the algorithms written by astronomers in the team do not scale well. All of these problems reflect common difficulties the 'average' astronomer† increasingly encounters as data-sets in astronomy grow.

Since the capacity of the personal laptop is no longer growing at the same pace as data-sets, other options must be made available to astronomers so that they may continue their research. One option is to train astronomers in the use of high-performance computing environments and parallelisation. However, such skill development is slow to permeate the community: even current PhD students often are not receiving the necessary training to efficiently handle the data-sets already extant. Another option is to provide analysis environments that are familiar to the general astronomer, but which can deliver the high performance necessary for large data-sets while minimizing new skills required.

† average astronomer: one who is not an expert in handling large data sets

New data archives that allow cloud-based computing, provide an exciting new possibility for allowing and even encouraging astronomers to analyze large data-sets. The work of the Australian Astronomical Observatory's Data Central project in this area is the subject of this talk.

2. Current Realities of Astronomy Analysis

Python is the *lingua franca*. In a recent survey, Momcheva & Tollerud found that 67% of astronomers regularly use Python. Python use is also growing in that a larger fraction of younger astronomers use the language than of older astronomers. For comparison, very few astronomers use SQL, the most common database query language—fewer, in fact, than use Microsoft Excel. Therefore, database querying, scripting and analysis should ideally all be possible in Python.

Data-sets are *large* and growing rapidly. Over its lifetime, the SAMI Survey may grow to approximately 20TB. However, projects like the Murchison Widefield Array already have data volumes in excess of 10PB—a thousand times more! The sheer size of these datasets make them difficult and expensive both to store and to move. Even moving something as small as SAMI is already time consuming. Therefore, it is critical that analysis platforms begin to offer the option of moving the code to the data.

Data-sets span many data centres. A single astronomical object might have detections residing in many different data centres, each with its own unique storage format and user interface. The standards introduced by the International Virtual Observatory Alliance provide consistent methods of accessing data from different data centres, as well as some measure of inter-operability between data centres. However, the increasingly multi-wavelength nature of astronomy research, and the goal of bringing code to data require a virtually seamless joining of remote data-sets within the analysis environment.

Astronomy is (embarrassingly) parallel. Many analysis work-flows involve executing large programs on each object of a data-set or matching a particular filter. As these programs are almost always independent of the results of other objects, this provides the perfect opportunity for parallelising execution with very little added complexity. Analysis on the 772 objects of the SAMI Galaxy Survey Release mentioned above could be run as 772 separate jobs on separate processors. Parallel programming, even for these simple cases, however, remains mysterious to the average astronomer. Therefore, analysis platforms should support intelligent, fully automatic parallelisation of such simple situations without any overhead required of the astronomer.

Documentation of the discovery workflow is becoming increasingly important. As analysis become ever more complex, the reproducability of results deemed so central to our scientific method becomes more elusive. Both understanding and repeating previous analysis is much easier when the original workflow has been clearly documented. Documenting the discovery workflow is easy with a notebook-like interfaces. These interfaces are not new (e.g. Mathematica), but do seem to be growing in popularity as more analysis platforms support them. Therefore, a notebook-like interface that is interactive and shareable is important for a modern analysis environment.

3. What is Data Central?

Given the current challenges faced by our users (such as the SAMI Team) and the realities of modern astronomy analysis, the Australian Astronomical Observatory (AAO) is developing "Data Central", an astronomical data archive with a long-term goal of supporting cloud-based processing and analysis. Initially, this archive will contain the

https://doi.org/10.1017/S1743921317001363 Published online by Cambridge University Press

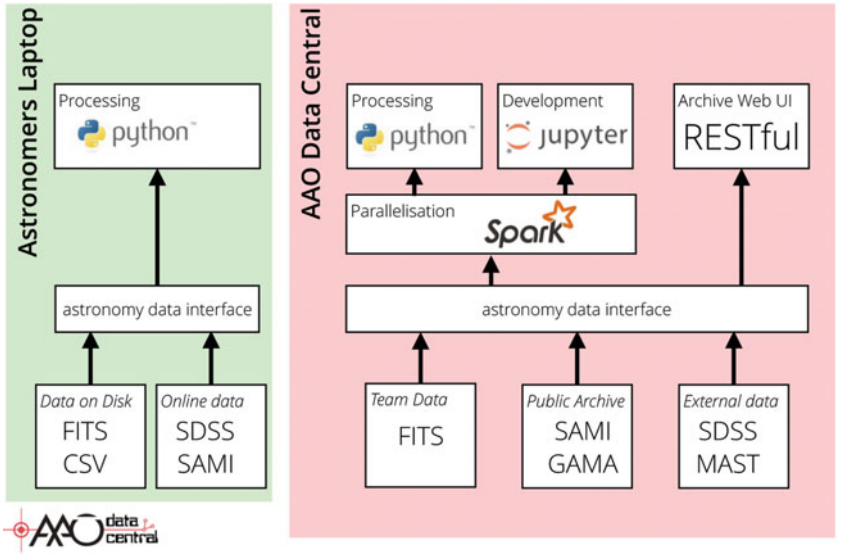

Figure 1. A systematic view of the Data Central system and the slightly modified analysis environment of the astronomer's laptop. An "astronomy data interface" layer abstracts analysis code from the data, making the code more portable between the astronomer's laptop and the Data Central system. In addition to the pure code-based analysis/development environment, Data Central also offers a notebook-like interface using Jupyter and a more traditional web-based query/explore/download system.

current SAMI Galaxy Survey data, and the recently completed GAMA Galaxy Survey. The inclusion of other major Australian optical data-sets, such as GALAH, OzDES, 2dFLens, TAIPAN and FunnelWeb is planned. These data-sets are relatively modest in size, enabling the project to experiment with cloud-based processing methods and corresponding user interfaces. As data-sets in Data Central are still accessible to more traditional analysis approaches, these experiments are largely free from time and design constraints.

4. Data Central Design Philosophy

Currently, astronomical analysis is most often conducted on the astronomer's laptop. Data is stored on the local hard-drive, and loaded directly into a numerical analysis package such as Python, IDL, or R. The astronomer may have collected the data herself, or have downloaded it from an online data archive.

Data Central's philosophy depends on a few changes to how the astronomer conducts analysis on her own laptop (left side of Figure 1). As the majority of astronomers use Python, the AAO has chosen to standardize on Python, and not support other languages. We are also introducing a data interface layer, which is intended to abstract details of the data storage (FITS, CSV, or remote data archives) from the implementation of the analysis in Python. This astronomy data interface makes the analysis code agnostic to the source of the data.

The structure of the analysis system at Data Central appears very similar to that on the astronomer's laptop (right side of Figure 1). Through the data interface layer, the astronomer may choose to access data public on Data Central's archive, data available from other astronomy data archives, or proprietary data uploaded to a secure area on Data Central. The same data interface layer means that analysis code written, tested

https://doi.org/10.1017/S1743921317001363 Published online by Cambridge University Press

and run on the astronomer's laptop can also be run on Data Central's servers without modification, even though the details of the data storage and access may be very different.

Data central will include functionality to parallelise execution of analysis code. Our goal is to make this parallelisation as automatic as possible. For analysis that executes independently on each object in a catalog—a situation we expect is very common—the parallelisation will be automatic and fully transparent. For more complex analysis operations, we hope to also offer easy-to-use tools enabling parallelisation while hiding the implementation details not relevant to the research astronomer.

To ease development and collaboration, we will offer a notebook-like interface. This interface will support a shareable, interactive, notebook-like workflow, most likely be provided by Jupyter, a web-based technology that runs inside a web browser. The notebook interface will complement an option for the astronomer to upload code developed on their laptop. We imagine that the astronomer might upload an analysis code, and then use the Jupyter interface to run that code across a catalog of astronomical objects and explore the results interactively.

Data central will also offer a web-based interface that is more typical of current data archives. The traditional interface will offer query, explore, and download functionality. A large part of this interface (query and download) will be built as a RESTful service, enabling programmatic access to the data online, as well as through the browser.

Finally, the AAO expects to offer users the option to scale their data analysis problems into the cloud. The AAO is not sure if it will always be able to support all of the demand for processing on Data Central in-house. Therefore, we would like to offer users with large analysis tasks the option to process their job on an external, cloud-based platform. To support this option, part of our architecture will be "containerised"—separated out such that it can be executed within a thin virtual computing host. Then, when a job's requirements exceed the quota of processing available to a user on the AAO's servers, that container can be moved into the cloud (see Figure 2). The Data Central system will take care of all of the details of collecting and moving code, data and results. This system will eliminate the need for the researcher to learn about and understand the many complexities and pitfalls of cloud-based computing. However, the cost for using a cloud-based system external to the AAO would still be borne by the astronomer.

5. Development to date

The AAO has funding to develop part of the evisiaged system over two years. Over the 18 months of that time already past, we have built much of the traditional part of the data archive: the web interface providing query, explore and download functionality. We have also been developing the data interface layer which will ultimately provide the abstraction necessary to make analysis code portable between an astronomers laptop, the Data Central servers, and the cloud. The team working on this system consists of about 2.5 full-time-equivalent-per-year effort spread between four developers (AWG, EM, LH, MV), plus three people providing management, applying for funding, and IT support (KS, SO, AMH).

6. Summary

In summary, we believe Cloud based Astronomical data processing should:

- be Python based,
- offer a shareable Juypter notebooks,
- support automatic and effective parallelisation,

https://doi.org/10.1017/S1743921317001363 Published online by Cambridge University Press

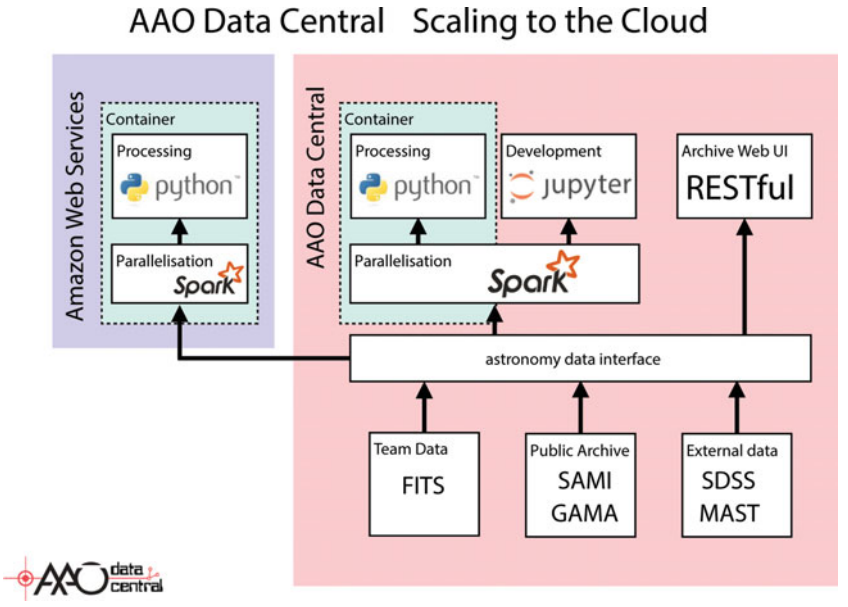

Figure 2. A sketch of how part of the Data Central system will be containerised for export to an external, cloud-based computing provider when the AAO cannot support the requirements of a particular analysis job.

- be abstracted from data storage in a portable way, and
- permit easy development in a familiar environment either online or on the astronomer's local computer.

The Australian Astronomical Observatory's Data Central is experimenting with these goals on modest data-sets now.

Reference

I. Momcheva & E. Tollerud. Software Use in Astronomy: an Informal Survey. *ArXiv e-prints*, July 2015.

https://doi.org/10.1017/S1743921317001363 Published online by Cambridge University Press