Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

Building LOFAR as a Service

A.P. Mechev,[1] J.B.R. Oonk,[2] A. Plaat,[3] A. Danezi,[2] and T.W. Shimwell[4]

[1]*Leiden Sterrewacht, Leiden, Zuid-Holland, Netherlands;*
apmechev@strw.leidenuniv.nl

[2]*SURFsara, Amsterdam, Noord-Holland, Netherlands*

[3]*LIACS, Leiden, Zuid-Holland, Netherlands*

[4]*ASTRON, Dwingeloo, Drenthe, Netherlands*

Abstract. The LOFAR radio telescope is a low-frequency aperture synthesis radio telescope with headquarters in the Netherlands and stations across Europe. As a general purpose telescope, LOFAR produces petabytes of data each year serving a wide range of science cases. The data volumes produced are difficult or impossible to process on a single machine or even a small cluster at a scientific institute. We provide a layout for serving LOFAR processing to the astronomical community by providing access to LOFAR pipelines accelerated on a high throughput platform. We build this on our previous success with parallelizing the LOFAR Surveys pipeline and with creating automated LOFAR workflows on a distributed architecture. The LOFAR As A Service platform will serve the LOFAR Key Science Projects (KSPs), specifically the LOFAR Surveys KSP, which aims to provide science ready products to the scientific community. Additionally, this system will provide a robust method to re-process LOFAR data with a single click.

1. LOFAR

The LOFAR radio telescope (van Haarlem 2013) consists of more than 7000 antennae with a dense set of core stations near Exloo in the Netherlands, 14 remote Dutch stations and a further 13 international stations across Europe. LOFAR is an aperture synthesis array operating at low frequencies, between 10MHz and 250MHz. Producing a science quality LOFAR image requires large amounts of processing to remove image artifacts produced by the telescope and the ionosphere (van Weeren 2016). The data reduction required to create a scientific image typically requires 256 GB of RAM and 4 TB of disk space. These resources in practice equal one or more dedicated nodes on a modern computational cluster. Requiring dedicated hardware for each data set makes it difficult to perform large scale studies using the resources provided by a typical research institution (e.g., a university). These processing requirements can only be met by using a large scale shared high throughput platform. We deploy LOFAR processing on three such data centers located at the LOFAR Long-Term Archive (LTA) sites. The LTA data centres are at SURFsara in Amsterdam, FZ-Jülich in Jülich and PSNC at Poznań with

the main deployment at SURFsara in the Netherlands[1]. Through this work, large LOFAR projects like the LOFAR Two Meter Sky Survey (LoTSS) (Shimwell 2017) can integrate their processing pipelines with heterogeneous infrastructure provided natively at each of the LOFAR Archive locations.

1.1. LOFAR Surveys

The goal of the LOFAR Two Meter Sky Survey is to make broadband radio images of the Northern Sky at an unprecedented sensitivity. The survey is expected to produce more than 3000 observations, each of which is more than 16 TB. While the raw data is stored at three Long-Term Archive locations, moving and processing this data at an university institution is untenable. To ease this issue, we have developed a framework, GRID_LRT, to parallel process LOFAR data at the SURFsara `gina` High Throughput Cluster (Mechev et al. 2017). A diagram of the processing steps for one of the LOFAR pipelines is shown in Figure 1. Using this framework, it is easy to efficiently process LOFAR data with various scripts, making it possible to create data products serving multiple science cases.

2. Pipeline Parallelization and AGLOW

Using the GRID_LRT software[2], we distribute LOFAR processing at the SURFsara site. Because of the data-level parallelism of LOFAR data, some of the steps can be run in parallel, thus helping accelerate the processing. While the parallelization helps accelerate a single run of the pipeline, it is still necessary to facilitate scheduling and running the pipeline automatically. To do this, we built a software suite to interface LOFAR workflows with the grid middleware and LOFAR Long-Term Archive. This software, AGLOW, allows users to build high-level workflows and easily process them on an European grid e-infrastructure (Mechev et al. 2018).

3. LOFAR As A Service with AGLOW

The AGLOW software is built on Apache Airflow[3] and has many options for triggering a workflow. Each AGLOW workflow can be scheduled automatically by the scheduler at a regular interval. Additionally, workflows can be triggered by another workflow or triggered manually by a user clicking the run button on the GUI or by the user launching the workflow through the CLI. Airflow also allows triggering of a workflow through a REST POST command, encoding parameters in the JSON payload.

Using this capability, we aim to integrate AGLOW with a lightweight web based front-end built on Django[4]. This front-end is responsible for authorizing the users, staging the requested data and passing the requested parameters to the respective AGLOW workflow. The front-end also presents the logs from the Grid jobs, AGLOW workflow

---

[1] `https://www.surf.nl/en/about-surf/subsidiaries/surfsara/`

[2] `https://github.com/apmechev/GRID_LRT`

[3] `https://airflow.apache.org/`

[4] `https://www.djangoproject.com/`

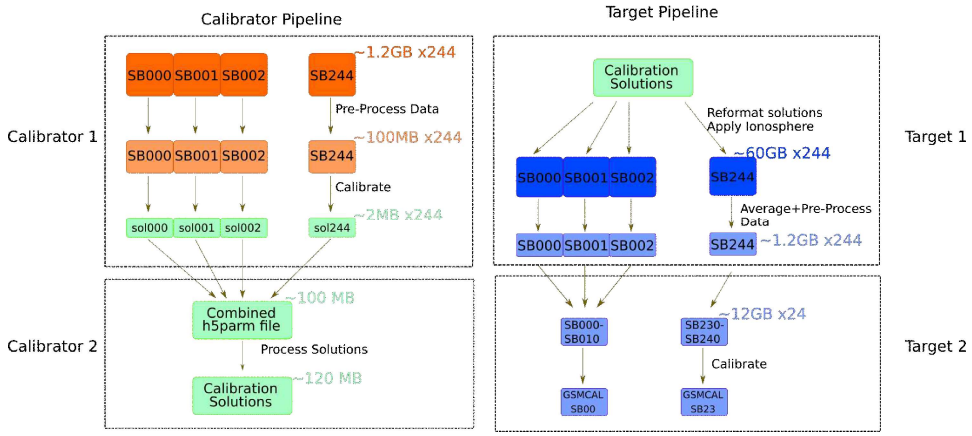

Figure 1. Parallelization for the LOFAR direction independent pipeline (de Gasperin et al. 2018).

and the front-end itself to the user. Additionally, these can be automatically sent to the user by email. Finally, the front-end can serve diagnostic plots to the users for each of their runs.

3.1. Adding and Modifying Pipelines

Each of the pipelines presented to the user contains a set of parameters. These parameters are sent to the AGLOW workflow and propagate to the relevant tasks. Ultimately, the worker nodes are fed the parameters and use them to process the requested data. Typical parameters are time and frequency averaging parameters which determine the time and frequency sampling rate of the final products. Other parameters include demixing sources, a string of sources that contaminate the observation and need to be explicitly removed. More complex pipelines can also request users to upload their own skymodels as text files, and use these skymodels to perform gain calibration on the requested data.

In order to add a new pipeline, the following steps need to be performed:

1. Define and test end-to-end pipeline execution at LTA sites
2. Decide on parameter types and values for pipeline steps
3. Implement pipeline in AGLOW including parameters (with default values)
4. Define JSON payload that will be sent by the front-end
5. Create a basic resource model describing processing time for different parameters (Typically data size, averaging parameters)
6. Add pipeline as an option to front-end

3.2. Performance Models

Processing LOFAR data can use significant computational resources. Presenting an easy to use interface to radio astronomers can lead to exhausting the requested resources at the institutions tasked with processing our data. Because of this, it is important to know roughly how many resources will be used by each job and prevent users from surpassing their allocated amount. As such, we aim to create simple processing models for each pipeline, and calibrate it as astronomers start using the service.

4. Conclusion and Future work

Creating a user-accessible portal for processing LOFAR data will be beneficial to the radio astronomy community, helping researchers efficiently process their data without having to dedicate computational resources at their universities, or spend time moving the raw data. This portal will include an authentication module, a workflow engine, and a performance model for each workflow. While the workflow engine exists and is currently in use by scientists, publishing such workflows to the broader community requires a model to predict the resources consumed by a single job based on the requested parameters. With such a model it will be possible to offer a managed service where the resources consumed by each project are measured and the job scheduling is optimized based on the remaining compute time and storage available on the shared infrastructure.

Acknowledgments. APM would like to acknowledge the support from the NWO/ DOME/ IBM programme "Big Bang Big Data: Innovating ICT as a Driver For Astronomy", project #628.002.001.

References

de Gasperin, F., Dijkema, T. J., Drabent, A., Mevius, M., Rafferty, D., van Weeren, R., Brüggen, M., Callingham, J. R., Emig, K. L., Heald, G., Intema, H. T., Morabito, L. K., Offringa, A. R., Oonk, R., Orrù, E., Röttgering, H., Sabater, J., Shimwell, T., Shulevski, A., & Williams, W. 2018, ArXiv e-prints, arXiv:1811.07954. 1811.07954

Mechev, A., Oonk, J. B. R., Danezi, A., Shimwell, T. W., Schrijvers, C., Intema, H., Plaat, A., & Rottgering, H. J. A. 2017, in Proceedings of the International Symposium on Grids and Clouds (ISGC) 2017, 2

Mechev, A. P., Oonk, J. B. R., Shimwell, T., Plaat, A., Intema, H. T., & Röttgering, H. J. A. 2018, ArXiv e-prints, arXiv:1808.10735. 1808.10735

Shimwell, T. W. e. a. 2017, A&A, 598, A104

van Haarlem, M. P. e. a. 2013, A&A, 556, A2. 1305.3550

van Weeren, R. J. e. a. 2016, The Astrophysical Journal Supplement Series, 223, 2