

Unlocking the LOFAR LTA

1st Hanno Spreeuw
The Netherlands eScience Center
 Amsterdam, The Netherlands
 h.spreeuw@esciencecenter.nl

2nd Souley Madougou
The Netherlands eScience Center
 Amsterdam, The Netherlands
 s.madougou@esciencecenter.nl

3th Ronald van Haren
The Netherlands eScience Center
 Amsterdam, The Netherlands
 r.vanharen@esciencecenter.nl

4rd Berend Weel
The Netherlands eScience Center
 Amsterdam, The Netherlands
 b.weel@esciencecenter.nl

5th Adam Belloum
The University of Amsterdam
 Amsterdam, The Netherlands
 a.s.z.belloum@uva.nl

6th Jason Maassen
The Netherlands eScience Center
 Amsterdam, The Netherlands
 j.maassen@esciencecenter.nl

Abstract—In this paper we discuss our efforts in “unlocking” the Long Term Archive (LTA) of the LOFAR radio telescope. This is a large (> 43 PB) archive that expands with about 7 PB per year by the ingestion of new observations. It consists of coarsely calibrated “visibilities”, i.e. correlations between signals from LOFAR stations. Currently, only a small fraction of the LOFAR LTA consists of sky maps, which are needed for most astronomical research. Unfortunately, creating such sky maps can be challenging, due to the data sizes of the observations and the complexity and compute requirements of the software involved. We try to fix this by enabling a simple one-click-reduction of LOFAR observations into sky maps for any user of this archive. This work was performed as part of the PROCESS¹ project, which aims to provide generalizable open source solutions for user friendly exascale data processing.

Index Terms—data dissemination, data aggregation, data mining, data processing

I. INTRODUCTION

The LOw Frequency Array (LOFAR), which covers frequencies between 10 and 250 MHz, is a European radio telescope with 51 stations that became operational in 2010. Its design differs from classical radio telescopes that consist of arrays of dishes. Instead, LOFAR combines the signals from 96 Low Band Antennas or 2×24 High Band Antennas (HBA field is split) into one station signal, similar to the signal of a single dish from a classical radio telescope. For imaging observations the station signals are correlated per baseline - a pair of two stations - following the principles of aperture synthesis; i.e. every pair of signals is multiplied and integrated over the time sampling interval. This means that a LOFAR imaging observation results in a large set of such correlations - generally termed visibilities. A visibility is recorded as a complex number for each baseline, frequency and time sampling interval and polarization product.

The LOFAR LTA [1] was set up to store all LOFAR observations. A typical LOFAR observation typically takes 8-12 hours and has a size of about 100 TB. Frequency averaging of every eight channels reduces this size to about

16 TB. Initial (coarse) calibration is also applied. The last few years the LTA has been expanding by about 7 PB per year and now (2019) it is exceeding 43 PB. These coarsely calibrated and frequency averaged observations are sets of visibilities that are generally not suitable as a starting point for scientific research. In astronomy, one generally embarks on a scientific investigation by inspecting sky maps. The motive may be, for instance, the detection of an explosion by a gamma-ray satellite at a particular position on the sky. Understanding the nature of such an event often requires multi-wavelength analysis. Consequently, astronomers will want to find out what kind of object images from observations at other wavelengths, like radio, show at this position on the sky. This is where a repository with low frequency radio maps covering a large part of the sky can provide a useful resource. Unfortunately, sky maps are available only for a small fraction of the LOFAR observations. This can be accommodated for by offering “one-click-processing” of observations: after selecting an observation, an appropriate reduction pipeline and a set of reduction parameters, acquiring a well calibrated sky map just requires waiting for staging - copying from magnetic tape to disk - and for processing to complete.

As mentioned above, the complex visibilities from the correlator are seldom used as a starting point for astronomical research. Instead, these data need to go through further processing such as radio frequency interference removal, calibration and finally imaging. These images can then be used as the beginnings of astronomical research. Each part of the above processing is complex and usually requires both domain and software knowledge in order to generate useful output. This, combined with the massive volumes of the data, makes the use of the LTA only possible for a select group of people. Furthermore, these challenges are exacerbated when one needs to process many observations. For this reason, researchers interested in surveys, for instance, are building frameworks to make LOFAR observation processing easy, portable, scalable and generalisable [2].

A large part of this research was conducted from one the

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement 777533.

¹<https://process-project.eu>

five use cases for PROCESS². One of the goals of PROCESS is to facilitate the use of the resulting platform by providing a user-friendly frontend that lets the user select both her dataset and workflow, launch the processing to the computing infrastructures and, finally, either directly or indirectly, retrieve the results from the frontend. For the particular case of LOFAR data processing, such a tool already exists, although it needs to be customised and extended; as part of the EU EOSC Pilot for LOFAR project³, an astronomer-friendly frontend to LOFAR LTA has been built.

A second goal of PROCESS is to offer exascale computing services to a range of scientists that require big storage and big compute facilities, in such a way that these users can remain completely agnostic of the location and specifications of the compute clusters where their data will be processed. In Europe, we currently do not have any exaflop supercomputer. This means that any form of processing requiring exascale computing will have to be distributed over a number of clusters in Europe. Such distributions over many clusters will have to be performed seamlessly and all software packages that run the computations will have to be containerized to guarantee portability. So, in summary, scalability is a main requirement for the PROCESS software infrastructure, not only with respect to compute power, but also with respect to data transfer.

One of the achievements of this project is to make workflow processing possible on a few clusters connected to the LTA - such as the clusters hosted by SURFsara⁴ - with high-bandwidth network and accelerating some of the processing steps by using parallel computing. This can be seen as vertical scalability. One of the goals the project did not reach yet is horizontal scalability. Indeed, PROCESS is designed to tackle the extreme large data handling and processing challenges introduced by exascale applications such as astronomical surveys. The approach taken by PROCESS is to build tools and services capable of supporting exascale applications by federating computing infrastructures from both HPC and Cloud.

II. IMPLEMENTATION AND RESULTS

A. Web Interface

The Web application (LTACAT) is a React application based on FRBCAT⁵ developed in the AA-ALERT⁶ project. This application is customised and extended to fit LOFAR LTA needs. In the frontend, the main database view (Fig. 1) remains mostly unchanged, but showing information about LOFAR observations, instead of fast radio bursts (FRBs). The frontend is however extended with a pipeline configurator and launcher. The backend is rewritten to use the LTA database view, which lets the application seamlessly access the observational archived data. Next, the backend is further extended with a RESTful [4] backend service developed using Django. The

²<https://www.process-project.eu/projects/square-kilometre-array-lofar-ska/>

³<https://eoscpilot.eu/lofar-data>

⁴<https://userinfo.surfsara.nl/>

⁵<http://www.frbcat.org>

⁶<https://www.esciencecenter.nl/project/aa-alert>

STARTTIME	ENDTIME	RIGHTASCENSION	DECLINATION	NR SUBBANDS
2012-03-09T13:28:00.000Z	2012-03-09T13:28:00.000Z	123.400216667	48.217295	80
2012-03-10T15:23:01.000Z	2012-03-10T15:23:01.000Z	65.7142858333	0	80
2012-03-10T11:50:12.000Z	2012-03-10T11:50:12.000Z	123.400216667	48.217295	80
2012-03-11T01:28:02.000Z	2012-03-11T01:28:02.000Z	183.103448333	23.08	80
2012-03-17T16:00:09.000Z	2012-03-17T16:00:09.000Z	123.400216667	48.217295	80
2012-03-18T06:03:01.000Z	2012-03-18T06:03:01.000Z	245.17241375	23.08	80

Fig. 1. Main database view.

Fig. 2. Prefactor listed as an available pipeline.

purpose of this service is to provide a window to the Web application to integrate and execute pipelines, i.e. it provides a list of available pipelines, provides configuration parameters for selected pipelines, and allows for executing a user configured pipeline for a selected observation. The pipeline configurator (Fig. 3) in the Web application is automatically rendered from the JSON schema⁷ that is provided by the backend service. JSON schema was chosen as the format for providing the pipeline configuration as it allows for, among others, setting default values and constraints on inputs, setting required properties, and defining dependencies between properties.

By default the backend service comes bundled with a pre-defined pipeline, but is easily extended. It allows users to develop additional pipelines and integrate them in the Web application. For this purpose a pipeline template is provided in addition to a step-by-step guide on how to integrate new pipelines into the service. The procedure consists of implementing a *run* function, defining the pipeline configuration parameters in JSON schema format, and registering the

⁷<https://json-schema.org/>

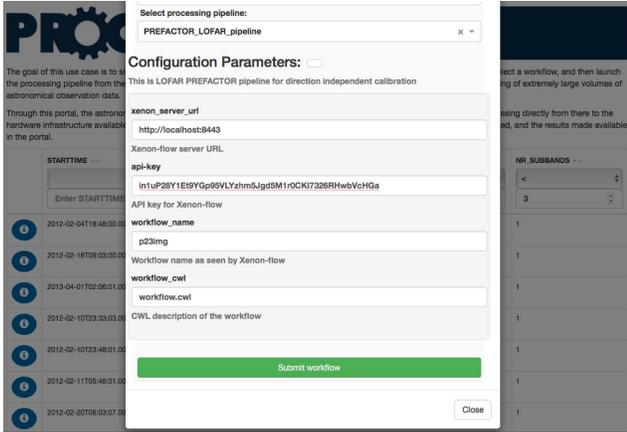


Fig. 3. Pipeline configurator.

pipeline in the pipeline administrator of the service. After installation of the new pipeline based on these steps, a new pipeline appears in the list of available pipelines next to the default one as illustrated in the image in Fig. 2. Here we added the `prefactor`⁸ pipeline.

B. DDF pipeline

`prefactor` takes care of the first step in calibration, which is direction independent. For science quality images, this has to be followed by direction dependent calibration. This is caused by the LOFAR station beam that covers a patch of the sky so wide that ionospheric phase delays differ within that patch. In order to take these effects into account the field of view is often split in hundreds of directions and calibration solutions are derived for each of these directions. A few calibration packages are available that solve for multiple directions on the sky, like `factor`⁹, `SAGECal`¹⁰, `BBS`¹¹ and `DPPP`¹², but we chose `killMS`¹³ because it is part of a complete imaging pipeline - the `DDF-pipeline`¹⁴ - that has proven its robustness in the the reduction of petabytes of LOFAR data as part of the LOFAR Two-metre Sky Survey [5]. The core part of the `DDF-pipeline` is a *self-calibration* loop: initial calibration solutions are used to produce a first map of the sky. The sources in that map are subsequently used to derive better calibration solutions, which are used to produce a more accurate map, and so forth, see [6].

C. CWL & Xenon

To run the pipeline in a scalable manner we chose to adopt the Common Workflow Language (CWL¹⁵) to describe the pipeline and use `xenon-flow`¹⁶ to run it on compute

⁸<https://www.astron.nl/citt/prefactor/>

⁹<https://github.com/lofar-astron/factor>

¹⁰<https://github.com/nlesc-dirac/sagecal>

¹¹<https://support.astron.nl/LOFARImagingCookbook/bbs.html>

¹²<https://support.astron.nl/LOFARImagingCookbook/dppp.html>

¹³<https://github.com/soapicc/killMS>

¹⁴<https://github.com/mhardcastle/ddf-pipeline>

¹⁵<https://www.commonwl.org/v1.0/>

¹⁶<https://github.com/xenon-middleware/xenon-flow>

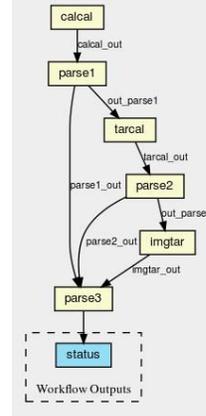


Fig. 4. Prefactor CWL workflow.

infrastructure. Xenon-flow is a RESTful Web service that executes CWL workflows on compute infrastructure using the Xenon [3] middleware and a CWL runner. It runs as a server in the background and waits for requests from clients.

As Xenon-flow runs CWL workflows, a CWL file describing the pipeline has been created. The implemented pipeline currently consists of three main components: a *calibrator*¹⁷ calibration, the target calibration and the imaging of the direction independent calibrated data. Each component is described in its own CWL file as a workflow step and wrapped in a container to ensure its cross platform compatibility. As the steps are dependent on each others output we parse the standard output of each of these steps for specific string values and have xenon-flow return the concatenation of these as final output status for the workflow. This leads to three more steps in the CWL workflow, each parsing the standard output of the previous main steps, the full DAG is shown in Fig. 4.

After clicking the “Submit Workflow” button, a HTTP POST request containing all required information for the pipeline is sent to xenon-flow server. Upon reception of the request, xenon-flow will start running the workflow. The configuration of xenon-flow will determine where and how the processes corresponding to the steps will be executed. In our case, xenon-flow is configured to run jobs on the Netherlands ASCI DAS5¹⁸ cluster located at the VU university using the CWL reference implementation *cwl-runner*. Since we are targeting an HPC compute cluster we use *Singularity* [7] as the container platform.

Upon completion, the concatenated string output is available in xenon-flow and is parsed. Following successful completion, FITS images produced by the imaging step are available on DAS5, from where they are retrieved using the Python *Fabric* module directly into the Django application directory structure. Once on the local machine, these FITS images are converted into JPEG images for visualisation in the browser using the Python *APLpy* module.

An example of images obtained through the developed pipeline is shown in Fig. 5.

¹⁷Radio source with known properties.

¹⁸<https://www.cs.vu.nl/das5/>

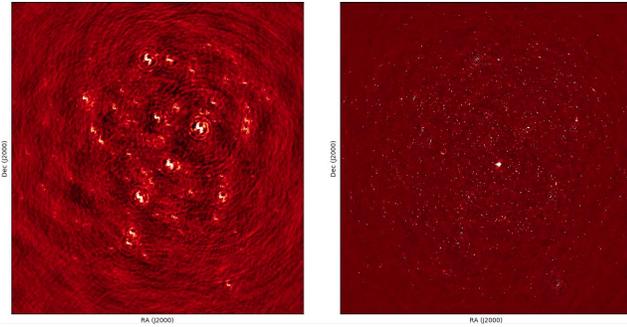


Fig. 5. Images of the patch of the sky covered by the processed data. On the left, the uncalibrated data and on the right the DI-calibrated data. DD calibration will further improve the quality of the image.

D. Data Services

The data infrastructure needed to unlock the LOFAR LTA has to be scalable, reliable, and easy to be installed and integrated which commonly uses data stack solutions by the scientific research community. The approach proposed to manage the data of the LOFAR LTA that can handle the multitude of different data models, applications, distribution and management is through virtualization, by encompassing all these requirements in a data micro-infrastructure with specific nodes for handling the different aspects e.g. data sharing (nextCloud) remote data access (GridFTP), and HDFS file system for computing, etc. The whole infrastructure then becomes an ensemble of micro-infrastructures each with its own full stack encapsulated in a virtual infrastructure. Through templating, micro-infrastructures can be booted up that will satisfy the groups' requirements for data processing. Cross provider data, process distribution and management are handled from within the micro-infrastructure. Cross group collaboration is also easily manageable e.g. a group could give access to another group through their data sharing node inside the micro-infrastructure. This decomposition allows for better scalability since state management such as indices, is split between many infrastructures. For this reason, we leverage the power of containers and created a platform using Kubernetes¹⁹ where users create and infrastructure with their own dedicated data services. Typical data services include data store adaptors to connect to remote data such as HPC file systems, native cloud storage using Ceph²⁰ block storage, runtime services that have access to storage such as WebDAV points, Jupyter notebooks and data staging services.

III. CONCLUSION AND FUTURE WORK

In this paper we described our ongoing efforts to develop a framework that enables simple one-click-reduction of LOFAR observations into sky maps. Ease of use and scalability are important requirements in the development of this framework. Our goal is to significantly reduce the time and effort required to generate skymaps from the selected observations in the

¹⁹<https://kubernetes.io>

²⁰<https://ceph.com>

archive. Currently, the framework allows for the user to select a LOFAR observation and a scientific workflow, launch the processing onto a (single) computing infrastructure and retrieve the results. The backend is developed as a RESTful service and allows for easy extension of the framework with additional pipelines. As a usecase the `prefactor` pipeline, a scientific workflow developed using both CWL and containerisation, was integrated into the framework.

While the current implementation fulfils the "ease-of-use" requirement, it is not yet sufficiently scalable. Currently, only a single compute infrastructure (DAS5) is available, and transferring data from the LTA over a regular internet connection is a significant bottleneck. In the coming months we aim to improve the scalability by making workflow processing possible on several clusters connected to the LTA through a high-bandwidth network, and accelerating some of the processing steps by using parallel computing.

For full calibration of LOFAR observations, the initial, direction independent calibration, using `prefactor`, is followed by direction dependent calibration using `killMS`. Direction independent calibration can be parallelized trivially by splitting the observation across subbands (sets of contiguous frequency channels) and distributing the fragments of the observation across the compute nodes of a cluster. For direction dependent calibration this is not possible because the signal-to-noise ratio from additional splitting of the sky into different calibration directions (facets) would become too low for finding stable calibration solutions. Therefore we currently perform this on a single high memory node. However, parallelization across nodes is still possible, e.g., by summing facets from different subbands to increase the signal-to-noise ratio. We are currently developing this code.

REFERENCES

- [1] G. A. Renting and H. A. Holties, "LOFAR Long Term Archive," in *Astronomical Data Analysis Software and Systems XX*, ser. Astronomical Society of the Pacific Conference Series, I. N. Evans, A. Accomazzi, D. J. Mink, and A. H. Rots, Eds., vol. 442, Jul. 2011, p. 49.
- [2] A. Mechev, J. Raymond Oonk, A. Danezi, T. Shimwell, C. Schrijvers, H. Intema, A. Plaat, and H. J.A. Rottgering, "An automated scalable framework for distributing radio astronomy processing across clusters and clouds," 12 2017, p. 002.
- [3] J. Maassen, S. Verhoeven, J. Borgdorff, N. Drost, cwmeijer, J. H. Spaaks, R. V. van Nieuwpoort, P. T. de Boer, and B. van Werkhoven, "Xenon: Xenon 1.1.0," Dec. 2015. [Online]. Available: <https://doi.org/10.5281/zenodo.35415>
- [4] R. T. Fielding, "REST: architectural styles and the design of network-based software architectures," Doctoral dissertation, University of California, Irvine, 2000. [Online]. Available: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [5] T. Shimwell, H. Röttgering, P. N. Best, W. Williams, T. Dijkema, F. De Gasperin, M. Hardcastle, G. Heald, D. Hoang, A. Horneffer *et al.*, "The lofar two-metre sky survey-i. survey description and preliminary data release," *Astronomy & Astrophysics*, vol. 598, p. A104, 2017.
- [6] C. Tasse, B. Hugo, M. Mirmont, O. Smirnov, M. Atemkeng, L. Bester, M. Hardcastle, R. Lakhoo, S. Perkins, and T. Shimwell, "Faceting for direction-dependent spectral deconvolution," *Astronomy & Astrophysics*, vol. 611, p. A87, 2018.
- [7] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLOS ONE*, vol. 12, no. 5, pp. 1–20, 05 2017. [Online]. Available: <https://doi.org/10.1371/journal.pone.0177459>