3rd URSI AT-AP-RASC, Gran Canaria, 29 May – 3 June 2022

Stimela2: Next-Generation Workflow Management for Radio Astronomy Pipelines

Oleg M. Smirnov*(1)(2) and Sphesihle Makhathini(3)(1)
(1) Centre for Radio Astronomy Techniques & Technologies, Dept. of Physics & Electronics, Rhodes University, Makhanda, South Africa, https://ratt.center
(2) South African Radio Astronomy Observatory, Cape Town, South Africa, https://sarao.ac.za
(3) School of Physics, University of the Witwatersrand, Johannessburg, South Africa, https://www.wits.ac.za/physics/

1 Extended Abstract

Stimela[1] was originally developed [1] as a *containerization* framework. It provided a standard set of Docker images (called *cabs*) for popular radio interferometry packages, a JSON-based schema specification language for describing the parameters of cabs, and a Python API for chaining cabs into *recipes* and executing the latter. Stimela enabled the creation of heterogenuous (i.e. mixing different packages, e.g. CASA, WSClean, MeqTrees, PyBDSF) and reproducible (run-anywhere, given Docker or Singularity container support) data pipelines. The CARACal pipeline (`https://caracal.readthedocs.io`) [2] uses Stimela as the underpinning technology, and has been adopted for and battle-tested with a number of MeerKAT projects, including the Fornax and MHONGOOSE LSPs.

Stimela2 represents a complete re-write of the framework, incorporating the lessons learned over a few years of using the first-generation product, and introducing a number of novel features. The goal of Stimela2 is to provide a complete framework for specifying massive distributed and reproducible data reduction pipelines, while supporting (at the pipeline component level) both legacy software and novel code. We will cover these features, and illustrate them with example recipes for very sophisticated and non-standard data reductions.

The original Stimela was focused on the stability and reproducibility of pipelines, by forcing all cabs (i.e. recipe steps) to be "official" container images. Stimela2 retains this capability, but at the same time also yields itself to "bleeding-edge" pipeline development and quick experiments. It does this by extending the definition of a cab to include native binaries (including ones called from within Python virtual environments), or even just importable Python functions. Stimela2 provides a unified interface to all three types of cabs, and allows them to be freely mixed within a data reduction recipe.

Stimela's strict up-front parameter validation has been retained, however the schema (and recipe) specification language is now based on YAML and standard Python type hints (https://docs.python.org/3/library/typing.html). This allows for the specification of fairly sophisticated schemas, and is able to seamlessly interface with a variety of command-line interface styles employed by different packages, while keeping the definitions concise and human-readable. The YAML stack is baced on OmegaConf (https://omegaconf.readthedocs.io) plus a number of custom extensions, which allows for composable, hierarchical and modular YAML files.

The recipe specification language now supports recipe logic such as for-loops and conditionals (in development), nested and composable recipes, and Python-style {}-substitutions for parameter values. This naturally yields itself to scatter-gather, and other kinds of distributed scheduling patterns. Work is in progress to allow Stimela2 to distribute recipes onto Slurm or Kubernetes clusters.

References

[1] S. Makhathini, "Advanced radio interferometric simulation and data reduction techniques," PhD thesis, Rhodes University, 2018. Available: https://vital.seals.ac.za/vital/access/manager/Repository/vital:26875

[2] G. Jozsa et al., "CARACal – The Containerized Automated Radio Astronomy Calibration Pipeline," in *Astronomical Data Analysis Software & Systems XXX*, ASP Conf. Series, 2021, in press

[1] *Stimela* is the isiZulu word for *train*.