Contribution ID: **57** Type: **not specified**

Framework for distributing Radio Astronomy processing across Clusters and Clouds

Friday, 10 March 2017 10:50 (30 minutes)

The Low Frequency Array (LOFAR) radio telescope stationed near Exloo, the Netherlands is an international aperture synthesis radio telescope developed to image the universe in the 20-200MHz frequency bands. Unlike telescopes using dishes or mirrors to focus light, aperture synthesis requires large amounts of processing between data acquisition and creating science ready images. While data can be split by frequency and processed in parallel, creating a wrapper around the processing software is needed in order to standardize data reduction across multiple locations while keeping track of the overall processing status.

This work presents a framework to wrap the radio processing software, a global location to track pipeline steps as well as an implementation of CERN's VM Filesystem (CVMFS) used to standardize the software install. With a central location to track the pipeline progress and a standard software install, this software suite allows to easily process LOFAR data on any computer, node or cluster that is connected to the internet. This distribution of processing allows to tap more resources than are available at any single cluster as well as a way to track the execution at all locations through a common interface.

Installing the LOFAR software on a CVMFS server allows every location to use the same software install, by tracking the software on the main server. The set of scripts that define the processing pipeline are placed in a sandbox folder along with a shell executable tasked with setting up the processing. The shell script sets up the environment, downloads, processes and uploads the data. The progress of the job is logged in an Apache CouchDB database inside a 'job token' document which contains all relevant data required to run the job.

The CouchDB token defines the parameters of the processing job. A user-friendly python interface was built to read/update fields from the tokens, download configuration files attached to the token, create and delete tokens as well as create/delete views from the database. Using this interface, it is easy to create batches of jobs to process large amounts of data on multiple nodes data centres or cloud services. Additionally, the python package logs the progress of each job by tracking the current processing step executed and updating the job token.

Further work will include integrating a scheduler which can make decisions which locations to use based on the current workload. Additionally, decision boundaries can be inserted between execution steps that analyse the intermediate solutions and decide on re-processing or terminating based on current data quality.

Primary author: Mr MECHEV, Alexandar (Sterrewacht Leiden)

Co-authors: DANEZI, A. (SURFsara); Dr OONK, Raymond (Leiden Observatory); SHIMWELL, T. W. (Leiden Observatory)

Presenter: Mr MECHEV, Alexandar (Sterrewacht Leiden)

Session Classification: Physics & Engineering II

Track Classification: Physics (including HEP) and Engineering Applications