URSI GASS 2021, Rome, Italy, 28 August - 4 September 2021

Towards the SKA science data processor

Peter Wortmann[(1)], Ben Mort [(1)], and Nick Rees[(1)]
(1) SKA Observatory, Macclesfield, UK, e-mail: {p.wortmann,b.mort,n.rees}@skatelescope.org

The SKA observatory [1] will be a prodigious generator of data. This is a major driving factor in the system design, where the Science Data Processors (SDP) will be operated as a schedulable resource integral to the operation of the observatory. The data challenge is considerable: Each SDP will consume a data stream of up to 700 GB/s from the Correlator-Beamformer and must provide sufficient computational power to reduce this to an average rate of approximately 10 GB/s total before distribution to a network of SKA Regional Centres. This data challenge, as well as the complexity needed to satisfy the evolving requirements of a wide range of science objectives, means that the development of new specialised software is a core part of the construction of the telescope.

SKA has adopted an open-source, lean-agile approach to software development. Guided by the SAFe™ framework, international teams of expert developers coordinate on three-month cadences and then follow a series of fast build-measure-learn cycles to develop and deliver on this plan. The architecture of the SKA SDP reflects these organisational and technical challenges: Our focus is building an open architecture for high-performance processing. This should allow us to absorb existing know-how from precursor telescopes and continue evolving as we learn more about operating the science facilities.

To this end, the SDP architecture [2] is split into layers, each with their own developer and user interfaces meant to enable frictionless evolution. The layers are:

1. Workflows, the primary "scripting" interface for parameterizing and coordinating processing blocks associated with an observation. They will determine what data products are generated and also enable top-level scaling involving "data island" partitions of the cluster.
2. Execution engine programs, which are instructions to execution frameworks on how to perform distributed processing to implement workflow steps. Internal data flows are likely to be the limiting factor to the SKA SDP scale, therefore we expect to be heavily iterating on fundamental technological choices in this space.
3. A processing function library that can be used to quickly build and modify radio astronomy pipelines in various execution frameworks. This separates algorithmic and domain-specific development. They could contain both functions adapted from existing software (especially precursors) as well as new and optimised kernels built specifically for SKA usage.
4. A standardised interface for high-performance data exchange between the mentioned layers. The SKA SDP is going to put significant effort into establishing portable in-memory data exchange technologies that can be used to communicate and share data between processing functions and execution frameworks with minimal copying involved. This will likely be achieved using techniques such as shared memory and portable in-memory data models such as Apache Arrow.

Our hope is that this technical and organisational framework will allow us to enable ongoing innovations in the radio astronomy data reduction field for many years to come.

References

[1] Dewdney, P.E., et al., 2019, "SKA1: Design Baseline Description", SKA-TEL-SKO-0001075

[2] P. Alexander, et al, 2019, "SKA1 SDP architecture", SKA-TEL-SDP-0000013