ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP072

OVERVIEW OF OBSERVATION PREPARATION AND SCHEDULING ON THE MeerKAT RADIO TELESCOPE

L.P. Williams, R. Schwartz, South African Radio Astronomy Observatory, Cape Town, South Africa

Abstract

The MeerKAT radio telescope performs a wide variety of scientific observations. Observation durations range from a few minutes, to many hours, and may form part of observing campaigns that span many weeks.

Static observation requirements, such as resources or array configuration, may be determined and verified months in advance. Other requirements however, such as atmospheric conditions, can only be verified hours before the planned observation event.

This wide variety of configuration, scheduling and control parameters are managed with features provided by the MeerKAT software.

The short term scheduling functionality has expanded from simple queues to support for automatic scheduling (queuing). To support long term schedule planning, the MeerKAT telescope includes an Observation Planning Tool which provides configuration checking as well as dry-run environments that can interact with the production system. Observations are atomized to support simpler specification, facilitating machine learning projects and more flexibility in scheduling around engineering and maintenance events.

This paper will provide an overview of observation specification, configuration, and scheduling on the MeerKAT telescope. The support for integration with engineering subsystems is also described. Engineering subsystems include User Supplied Equipment (USE) which are hardware and computing resources integrated to expand the MeerKAT telescope's capabilities.

OBSERVATION SPECIFICATION

An observation is a general term for any activity that is performed using the combined telescope systems. Observations are loosely divided in ***science*** and ***engineering***. Science observations begin with a *proposal*. The proposal describes the science case/motivation, in addition to certain key requirements that may be fulfilled by the telescope. This initial information is described in a *proposal* submission. A proposal is a formal document that describes the full observation campaign including: the scientific merit, the required observing time and the telescope features/functionality required.

The technical feasibility of the proposal is assessed by the resident astronomers who are familiar with the telescope systems and capabilities. These individuals are able to translate the required telescope functionalities into the real telescope subsystems and resources that enable the required functionalities.

If the proposal is accepted, then the parameters are captured into a set that will enable configuration. The MeerKAT telescope [1] has a particular set of proposals that will consume a majority of the observing time with only a smaller fraction being reserved for other proposals awarded at the director's discretion.

Engineering observations are generally shorter activities. The planning and scheduling of engineering observations is done directly with the Telescope Operations team, but uses most of the same mechanisms and data structures used for science observations.

OBSERVATION CONFIGURATION

The observation configuration (also referred to as *experiment configuration*) on the MeerKAT telescope, inherits from the operational learnings of the Karoo Array Telescope (KAT-7) [2]. The KAT-7 telescope was composed of 7 antennas operating as a single array. It was operated 24 hours a day, 7 days a week with ad hoc maintenance events in daylight hours of a work week. This predecessor instrument experienced a relatively brief time as an operational telescope servicing a variety of engineering and scientific observations. For MeerKAT the two types of observations, engineering and science, are described further.

Engineering Observations

Engineering observations are requested by engineers, technicians and science commissioners to characterise, test and monitor the telescope and its subsystems. The periods of engineering observations are defined in local time and they will require specific components to be exercised. This may or may not entail movement of the antennas in the array.

Real time sensor data is of prime interest to system engineers, rather than data collected from the receivers with some exceptions. Sensor data is used to assess hardware performance, predict failures and inform development of new features. The data set of interest is encapsulated by the period of the observation and contains sufficient information for the requesting party to continue their work.

Science Observations

Science observations are requested by science commissioners and external scientists. These observations require specific movement patterns on or around sky coordinates. The observing periods are defined in sidereal time and will utilise specific telescope resources. Scientific data captured from the receivers is of prime interest but sensor data is also required for calibration purposes.

The astronomical sources which need to be observed are not always above the horizon. To gather sufficient data, the same observation would need to be repeated on different

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP072

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

days at the same sidereal time, forming a campaign that may last many weeks.

The commonality between engineering and science observations can be simplified as:

- A set of telescope components/resources required (this abstracts both hardware and software items)
- A set of sky coordinates to be observed (which may be 1 or even none for engineering exercises such as drive motor tests)
- A telescope pointing/movement pattern on, around or between the provided sky coordinates (which may be static azimuth and elevation)

The above set of common observation features are collected into a data structure known as a *schedule block.* The schedule block also includes meta data such as the individual/group that requested it, and execution constraints that can be defined by the system, Operations or the requester. The schedule block controls telescope movement and resources via a prewritten script, created using the Python programming language.

Schedule blocks may be collected into larger structures called *program blocks*. Program blocks store meta data and constraints for managing the science observation campaigns which span multiple days over the course of weeks and even months.

Program Blocks

Accepted proposals may be defined as program blocks (PBs) composed of one or more schedule blocks (SBs). Most proposals are defined in a single PB defining its complete execution in a set of SBs. PBs can also be used to divide up the observation time of a proposal e.g. a PB can be defined for each month of a long running proposal. PBs are DEFINED when they are created and the required fields populated. They become APPROVED once their contents have been verified by staff scientists and the Operations Team. Below is an excerpt of parameters available in the PB data structure.

1. desired_lst_start_time: a sidereal time window that corresponds to a consistent position of interest in the sky
2. zones_of_avoidance: areas of sky that telescope should avoid pointing to due to proximity to interference such as satellites or the sun
3. obs_constraints: constraints related to hardware availability such as antenna baselines or environmental conditions such humidity
4. tac_priority: time allocation committee (TAC) assigned priority based on the science proposal, scheduling and other factors

The MeerKAT *Control and Monitoring System* (CAM) gains scope of PBs and SBs that are APPROVED, and they are then available for scheduling and execution on the active telescope subarrays. Subarrays are controllable groups antennas and signal path resources. Subarrays may be composed of all available MeerKAT antennas or a select subset.

Schedule Blocks

A Schedule Block (SB) is the atomic unit of execution in the MeerKAT system. It cannot be paused and resumed, but can be stopped or cancelled. SBs are created similarly to PBs. APPROVED SBs (as with APPROVED PBs) become available for scheduling and execution on the active telescope subarrays. Below is an excerpt of parameters available in the SB data structure.

1. verification_state: a state flag updated as the SB is run through a 'dry-run'. Dry-runs are simulated executions of the SB and employ simulated systems.
2. obs_readiness: a flag calculated by testing the observation constraints against the system state and environmental sensors
3. instruction_set: list of command instruction predominantly for the antenna positioners but may include command to engage features of other resources required by the observation
4. controlled_resources_spec: defines a list of hardware/ software elements that form the telescope signal path and data capture
5. config_label: a hash label capturing the state of deployed telescope configuration at the time that the SB is executed
6. expected_duration_seconds: the expected duration as calculated by the 'dry-run' simulation
7. horizon: the lower elevation limit to which the telescope will track a celestial target as it traverses the sky
8. queue_position: the position in an execution queue as calculated by the automatic scheduler

The PB and SB data structures span two related problem spaces; long term scheduling and short term scheduling.

LONG TERM SCHEDULING

For science observations, the parameters are known well ahead of time. The sky elevations at specific sidereal times can be calculated and scientists are able to submit the required amount of observing time to gather sufficient signal energy from the sources of interest. The scientist submitting a proposal is referred to as the *primary investigator* (PI). There may be many other collaborating scientists associated with any given proposal.

Once a proposal is accepted, the requested observation and key instrument parameters are captured. An approved proposal will receive a unique proposal ID and a total time allocation which will be broken into several time increments defined as schedule blocks that fit the daily use of the telescope.

Proposed observations may run over several days, weeks or even months, e.g, *Proposal "X" has been allocated 40 hours of observing time and the data is collected 8 hours per day over a 5 day period.*

To facilitate planning, capturing and testing an approved observation proposal, scientists make use of the MeerKAT *Observation Planning Tool*.

The collection of available program blocks are populated with schedule blocks which are then used for composing a

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP072

schedule for the week. This schedule is formulated by an *astronomer on duty*, a scientist with broad knowledge of the science proposals and who works with the Operations team.

Observation Planning Tool

The Observation Planning Tool (OPT) is a visual tool for planning MeerKAT observations. It is available on the public internet for anyone that has a Proposal ID to use. Its functions include:

- Defining appropriate schedule blocks for proposals
- Populating the potential observation schedule for the telescope
- Simulating execution of planned observations
- Reporting on the completion status of observations
- Notifying users of progress and outcome of the observation

The desired telescope and correlator configuration, start/end times, target and calibrator sources are all specified via the OPT during planning. OPT allows users to simulate planned observations with or without a telescope setup, providing graphs and readable tabular data of simulation results. OPT can then create schedule blocks corresponding to the planned observation through the MeerKAT *Control and Monitoring* (CAM) system. Each schedule block has a unique ID code and contains various configuration data, e.g: proposal ID, owner, type, description, controlled resources, desired start time, etc.

OPT is also able to generate google calendar event entries to keep astronomers informed and assist telescope operators to prepare the necessary telescope subarray configurations.

Schedule blocks are eventually executed by telescope operators using the CAM system at the appropriate time via the *Kat-scheduler* process.

SHORT TERM SCHEDULING

Short term scheduling is managed by staff astronomers and the telescope Operations Team. They use a CAM system subprocess known as *Kat-scheduler*, which is the main manager of SB states and is responsible for updating those states in the Obs Database. Kat-Scheduler monitors the execution state and progress of a running SBs, and updates the SB state parameter accordingly. While it is referred to as a scheduling tool, this process operates as an execution queue, ordering and running SBs

Kat-scheduler enables Operators to perform dry-run simulations of SBs before execution. The dry-run verification step is used to:

- Check that schedule block instruction-set scripts have correct parsing and syntax
- Calculate the expected execution time (duration) of an observation
- Generate a readable list of all the sky targets that the observation will visit over the duration of the schedule block

The MeerKAT telescope is able to activate and manage up to 6 independent subarrays. Each subarray on MeerKAT has its own "*sub-scheduler*". Kat-scheduler provides a single control interface for all sub-schedulers and routes requests using the relevant subarray.

Scheduler Queue Processing Cycle

1. Process schedule on each sub-scheduler in sequence.
2. Update SB "readiness" by calculating constraints (not in zone of avoidance, are resources available, is target up, etc.)
3. Manage queue behaviour based on scheduler mode (QUEUE, AUTO, etc.)

Each sub-scheduler is completely independent of the others and only operates on resources and SBs assigned to the associated subarray.

Scheduler Modes (per Sub-Scheduler)

Operators control and monitor the sub-schedulers via a graphical user interface. The interface is built atop an older, console-based interface which may be used in special circumstances. Operators can set the sub-scheduler into any of the following modes:

1. Idle: Sub-scheduler starts in this mode but then immediately transitions to one of the other modes.
2. Manual: Execute SBs in the queue only when user manually commands it.
3. Queue: Execute SBs in the queue in the order they were added to the queue by Operators. Execution will continue until the queue is exhausted.
4. Locked: Immediately prevent sub-scheduler from further execution of SBs. This state may be activated manually by Operator intervention or autonomously by the CAM system if an alarm condition is detected.
5. Auto: Execute SBs in the queue in a dynamically adjusted order calculated by a ranking function. The order can change without Operator intervention and execution continues until the queue is exhausted.

Sub-Scheduler and Schedule Block Types

Different SB "types" interact with the sub-scheduler queue in different ways:

Observation SBs are the main type of schedule block used for science. This type of SB will end automatically when the schedule block instruction-set script reaches its conclusion and thereby leave the execution queue.

Manual SBs do not generate task execution logs. These SBs occupy and control the required telescope resources as defined by their "controlled resources" parameters, but do not immediately use them. Operators are able to micro-manage the SB resources via the same internal interfaces that would normally be used. This type of SB is used for engineering and has to be manually stopped by an Operator.

Maintenance SBs are similar to the manual type, but typically used to denote a maintenance task, e.g. lubricating vacuum pumps. This is useful for monitoring purposes as resources under maintenance will often have sensor states that can raise alarm conditions if not clearly flagged. This type of SB has to be manually stopped by an Operator.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP072

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Sub-Scheduler Functionality

The sub-scheduler is required to process the execution queue as an *observation schedule* and select ready SBs for execution. It enables Operators to validate APPROVED SBs by determining the availability of resources required for SB and calculate whether the observation constraints have been met. Before the real execution of a SB, another dry-run simulation is always done to determine if all temporal parameters are still valid and obtain an up-to-date estimation of total execution time.

The sub-scheduler will re-compute the execution queue at the following events:

1. When a SB finishes, ie. completes or is interrupted (as the available system resources will be updated and affect the readiness of SBs)
2. After a SB is submitted for execution (as the available system resources will be updated and affect the readiness of SBs)
3. Periodically every 5 minutes after completing the previous scheduling cycle (to re-evaluate system resource availability and observation constraints)

Automatic Scheduler

The *automatic scheduler* (AS) is engaged when setting the sub-scheduler mode to '***auto***'. The auto scheduler will only attempt to rank schedule blocks that have been assigned to the subarray. It uses a ranking function which calculates a score for individual SBs, determining their position in the execution queue.

Astronomical observation scheduling optimization is described as an NP-hard optimization problem [3], and may be computed using integer linear programming (ILP) or mixed integer linear programming (MILP). Thus a formula containing weighted ranking factors is created to dynamically calculate a dynamic SB execution queue at regular intervals or triggered by certain events.

The Meerkat AS is experimental and uses a best guess of the weights assigned to the different factors that should be adjusted as the functionality of the AS matures. Simulation is used to analyse the impact of different weights and calculate the initial values of the ranking factors.

The ranking function is as follows:

$$Rank = 1000SRC + (25(OC_1 + OC_2) + 10OC_3 + 40OC_4) + OE + 1.2TAC + 0.001UDP + 1.05PNC$$

Where:

SRC: System resource constraint (e.g. required antenna groups within the subarray)

OC_1: Sources to observe are above the horizon limit

OC_2: Tracking through zones of avoidance

OC_3: Sensor value constraint (e.g. humidity weather sensor reading)

OC_4: Time window for execution (e.g. sidereal time range for optimal observing)

OE: Observation efficiency calculated as time spent transitioning between sky targets defined in the SB instruction set

TAC: Time Allocation Committee priority score

UDP: User defined priority allowing ranking importance of SBs within their associated PB

PNC: Program nearing completion indicating how much data out of the total required has been collected for the parent proposal

The MeerKAT AS functionality encompasses a basic framework to compute the rank for SBs and to specify constraints for SBs. Only a limited number of ranking factors have been implemented in the initial version. Some constraints already exist on PBs and SBs, for example; sensor values expressions, resources required, desired LST start time, desired start time, etc.. However, the framework does support the inclusion of any future parameters that may be defined.

The MeerKAT AS also includes a basic simulation framework to set the weightings of ranking factors and evaluate the effectiveness of the changes in the weighting of ranking factors. The system provides a rich test bed for various machine learning and data science projects relating to scheduling and telescope usage.

User Supplied Equipment

User Supplied Equipment (USE) are hardware elements that are installed on the MeerKAT telescope to expand its functionality. Institutions fund the development, construction and integration of these systems.

Within the CAM environment, these instruments become resources that are available for PBs and SBs to list as required. As such they can impact short term scheduling via constraints based on their operational state. USE instruments include:

- S-band receivers
- Poly-phase filter-bank and correlator
- Pulsar detectors
- Computational systems searching for indicators of extra-terrestrial life

CONCLUSION

The Meerkat Telescope incorporates a wide range and features and systems to facilitate efficient scheduling. Effective and cost-saving use of telescope time is of great importance, especially to the Operations Team.

New feature requests and requirements are always emerging from astronomers and engineers. While great strides have been made, there are still many improvements underway.

Within the engineering and software teams there is also great interest in data science using telescope activity and sensor data, as well as applying the latest techniques in machine learning. This will facilitate moving towards more automation within the various observation scheduling and queue management systems.

REFERENCES

[1] MeerKAT Array Release 1, Information Sheet, https://www.sarao.ac.za/wp-content/uploads/2016/07/info_sheet_ar1_2016.pdf

[2] KAT-7, http://public.ska.ac.za/kat-7

[3] A. I. Gómez de Castro and J. Yáñez, "Optimization of telescope scheduling", *A&A*, vol. 403, pp. 357-367, 2003. doi:10.1051/0004-6361:20030319