

VLA data sender description

This app repository is at https://gitlab.nrao.edu/nms-evla/nrao-ods/vla_data_sender . Its README contains the history of changes.

Summary

- Background application that checks for new observations frequently. Cycle time of the application: 01 minute (can be adjusted as needed).
- Initial app language: Python.
- It "guesses" the next target by checking the latest schedule created by the OST. The targets' data (eg., coordinates, frequencies) are retrieved by a query to the VLA Observations DB.
- It confirms the guest by checking ``.evla` scripts generated by the OST. It assumes that the most recent one refers to the current or the next observation, based on its time contents.
- The app handles multi-band (sub-arrays) and simultaneous observations (eg., VLA+VLBA).

Justification

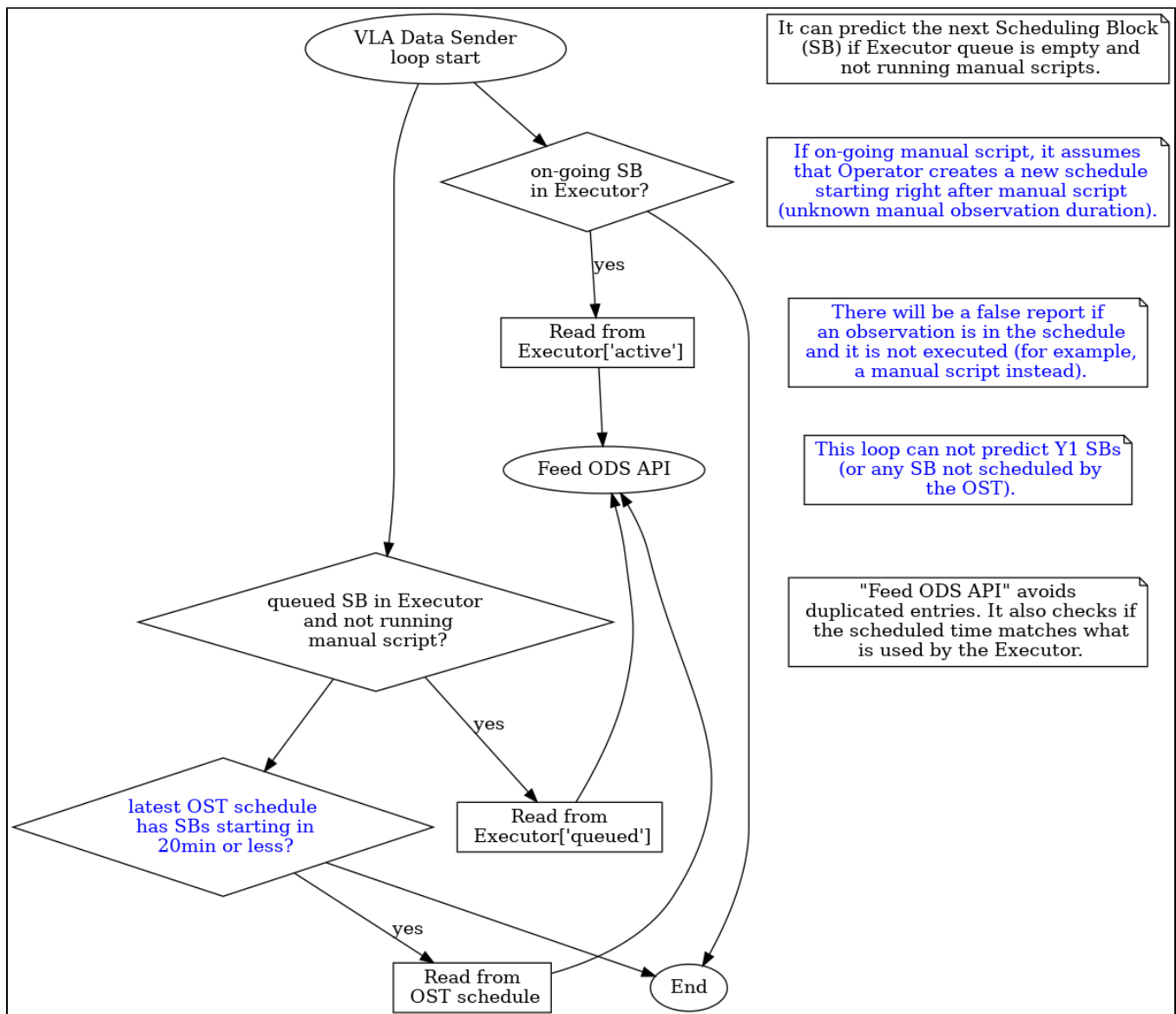
The information needed to feed the Operational Data Sharing DB (ODS DB) are originally stored in the (postgres) DB at `sssprowdb.aoc.nrao.edu` ("chewbacca" machine). When the Operator creates a schedule and accepts a SB, the OST reads that DB and create the ``.evla` and ``.vci` scripts (to the Executor and correlator, respectively).

It would be easier to implement an application that only reads these script files and sends them to the ODS DB. However, the Operators can accept the next SB just a few seconds before it starts. Thus, in order to meet the 20 minutes in advance requirement, the feeder application needs to constantly monitor for new schedules created by the OST and **confirm** that the next starting target from the schedule was effectively accepted by the Operator.

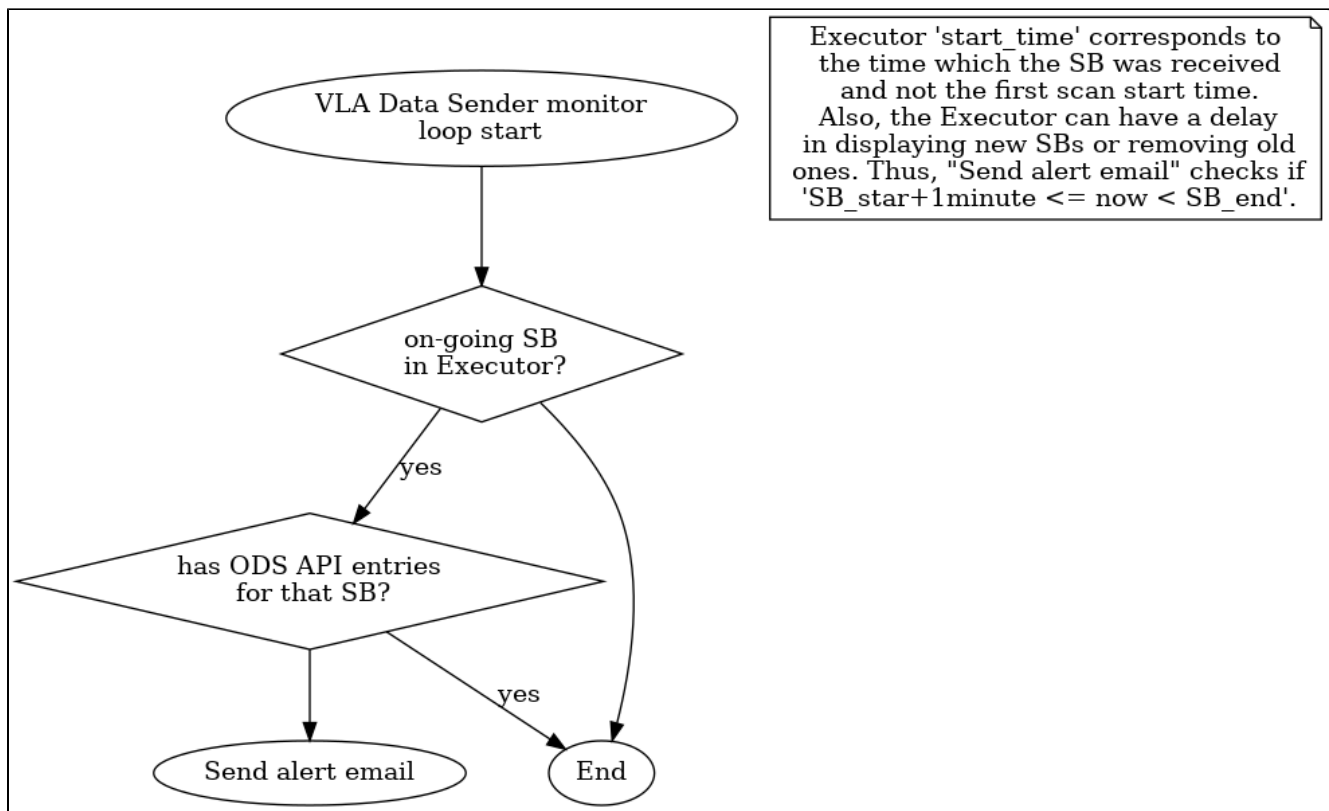
The VLA Data Sender could be integrated in the OST. However, it is complex to make changes to OST, and it is a good feature to be able to be used if manual scripts are executed (outside the Dynamic Scheduling mode).

App flow description

VLA Data Sender:



VLA Data Sender Monitor:



JSON keys

The JSON keys are detailed in the ODS API documentation. More information at [ODS Database API Operation Description](#).

Current Limitations/ToDo

Version 0.6 of the Data Sender is the "Hello World" version. Versions 0.7+ are for the post "Hello World" phase (Fall 2023 - Spring 2024).

- Version 0.6 only reports `is_pulsar_mode_bool`. No integration times.
- The data sender doesn't handle "on source" scans properly (short timestamp lengths).

Version 0.7 released on Nov 1st, 2023:

- It now reports the "integration time" (dump length) for regular observations (not for Pulsar ones).
- SB scans can be defined in terms of "on source" time (versus the default "duration").
- After fixing "on source" scans, all known discrepancies in the different timestamps are smaller than 5 sec (between this data sender, schedule = OST and execution = Executor).
- It now reports the "slew_sec" in the JSON as the slew time in seconds (i.e., starting at the "src_start" timestamp, after the slew time the array should be in the "on-source" position).

Version 0.8, release date **TBD**:

- It reports the "integration time" (dump length) for all observations, including all Pulsar modes.

Future versions:

- Ensure that no sources are using 1950 epoch (instead of 2000)
- Support "holography" scans
- A logic/mechanism to observation updates is TBD (eg., new start+end times from the same obs_id), to avoid duplicates or outdated information.
- A web-based GUI (see "GUI description" below).

Notes

- Monitor the OST-to-Executor commands or the Executor broadcast messages to guarantee that the next target guess was made correctly. An answer: this shouldn't be needed, as the *.evla scripts should contain all the info needed.
- Integrate this feeder inside the OST. An answer: it is a good feature to be independent of the Dynamic Scheduling.

- Coordinates J2000 or current? Proper motion explicitly or not? An answer: J2000, with retrospective proper motions taken into account when needed.
- Angles are defined in degrees (including Right Ascension). Although radians is the default unit for angles in the international system, ISO 6709 sets degrees for point location by coordinates. Also, major astronomical tools (Simbad, AstroPy) use degree instead of radians.

GUI description

- A way to stop the data sender when the VLA is not in Dynamic Scheduling mode (i.e., when OST is not in use). An answer: Operator GUI.
- It informs the app status: "green" while the end-time of the last DB update if longer than 20 min from current time. "red" status otherwise.
- It informs the last DB update time.
- It has a button to stop the updates ("gray" status).
- It has a form to send manual ODS-DB updates (TBC).

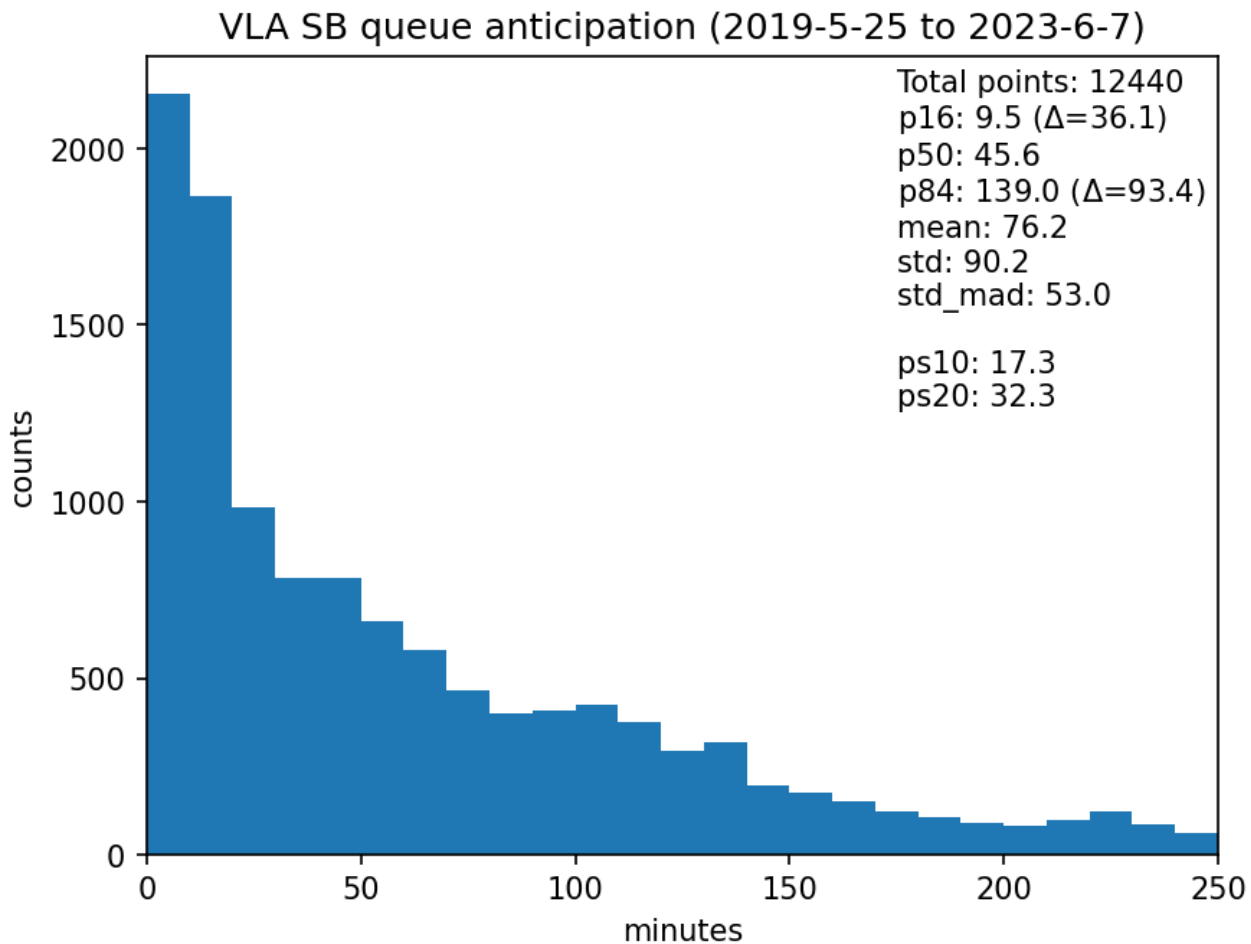
Acronyms

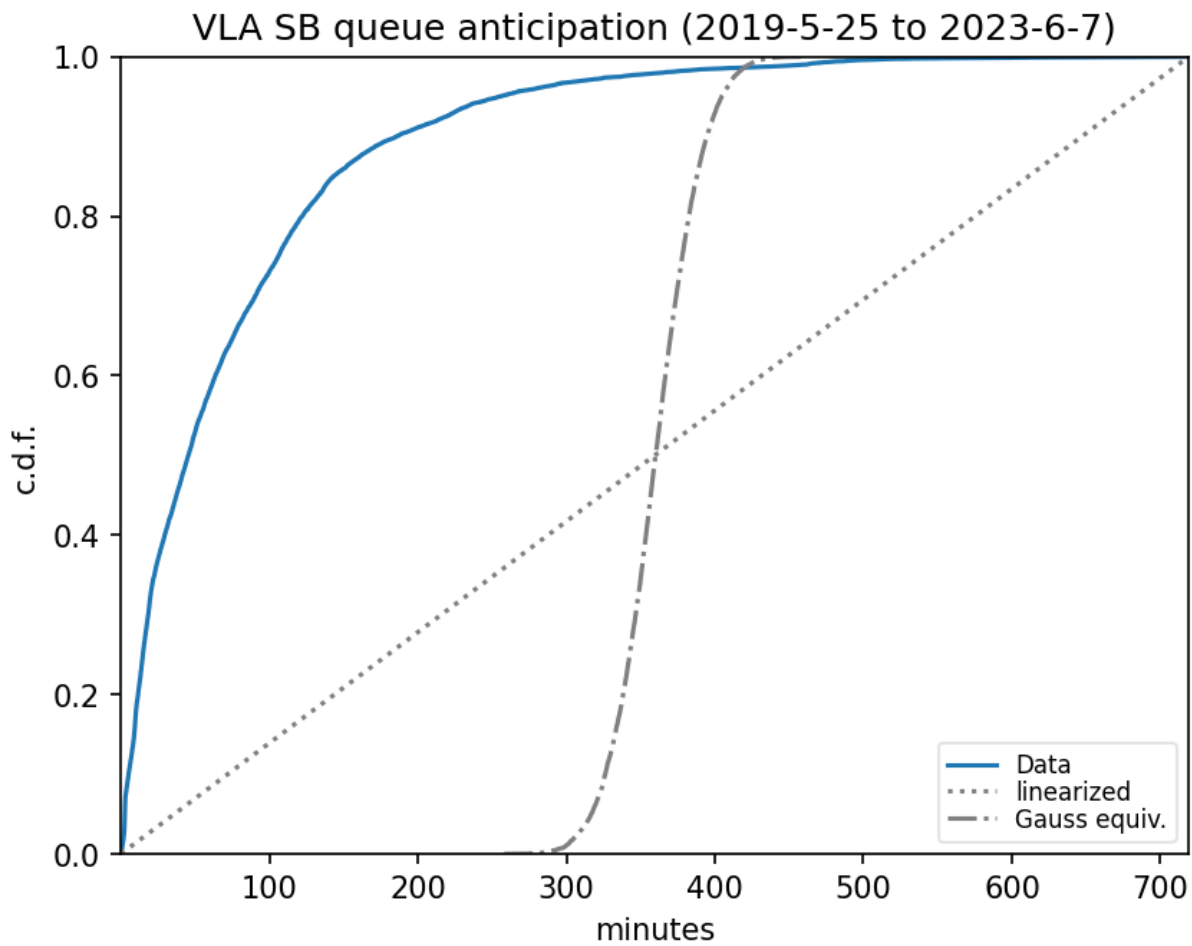
- DB: database
- ODS: Operational Data Sharing
- OST: Observation Scheduling Tool
- SB: Scheduling Block

SB queue anticipation

In the last four years (mid 2023 -- mid 2019):

- 67.7% of the SBs were set more than 20 minutes in advance;
- 32.3% of the SBs were defined in up to 20 minutes;
- 17.3% of the SBs were defined in up to 10 minutes.





One could argue that the satellites configuration time could be accommodated in the gaps the dynamic scheduling creates (the major driver of the gaps is to accommodate priority observations.) I don't have the length distribution of gaps (yet), but I recently found that they account for less than 3% of the dynamical time of the array. The typical SB length is 2 h = 120 min. 3% of that is 3.6 min. Thus, as long the satellite configuration time is longer than that, we would have impact on the array operations.