# God of schedulers

Co-learnium

Vivek Gupta

09/03/2023

# UTMOST

1.6 km interferometer; 2 arms

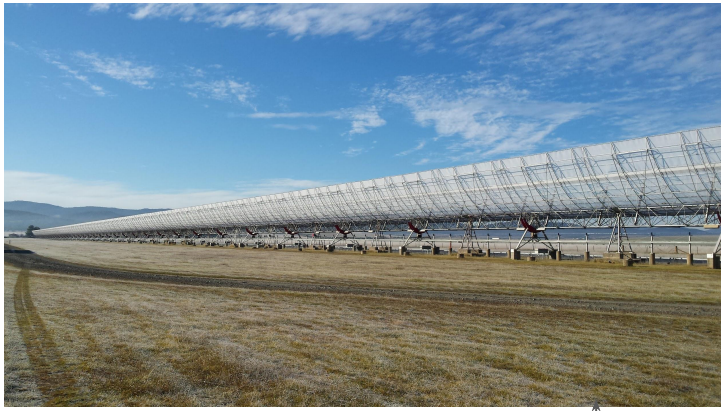
E-W, N-S operated independently

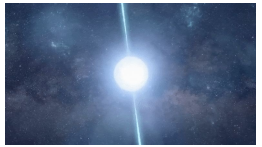
Transit-only instrument - phased array config

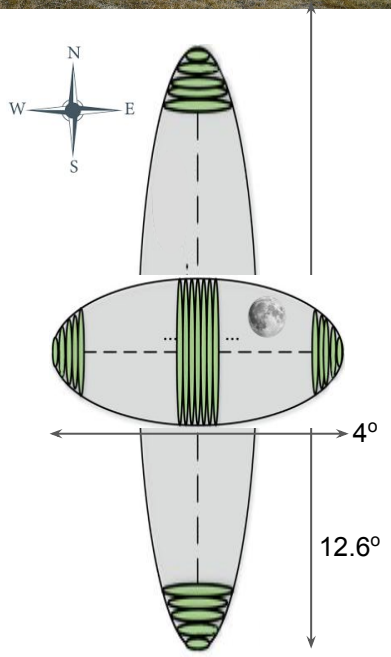
FoV filled with 100s of narrow fan-beams

Can steer in NS mechanically/electronically

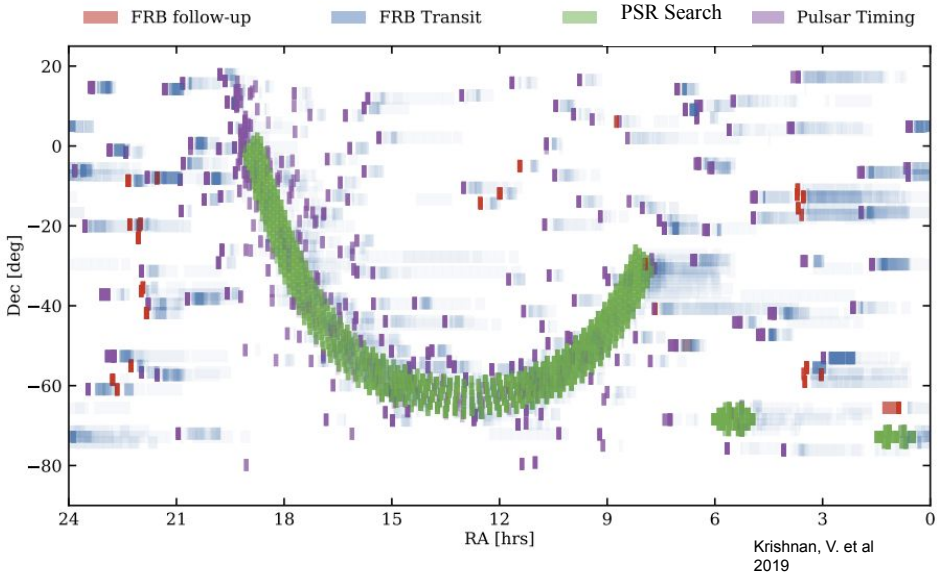
Designed to run real-time FRB/PSR search and pulsar timing programmes simultaneously [~17 FRB detections, ~400 pulsars timed]







# Typical Observing strategy




Steps to carry out an observation:

1. Wait for the field to reach zenith
2. Slew the telescope to the desired coordinates
3. Trigger the backend to start recording in appropriate mode - FB / Fold / FB + Fold / CORR
4. Stop recording after target exits the FoV

\+ 1 phase calibration obs per day

# Initial schedulers

Fabian

Aditya

**AUTOMATIC MODE**

Static schedule based software – similar to PKS/ATCA.

User specifies obs parameters in a file, which is executed linearly.

```
[vgupta@mpsr-srv0 ~]$ cat schedule.sch
POSN; 00:00:00; -28:06:00
WAIT; LST; 03:20:30
fb; transiting; CDFS; 03:30:00; -28:06:00; 1200
```

```
[observer@mpsr-srv0 ~/schedule]$
[observer@mpsr-srv0 ~/schedule]$
[observer@mpsr-srv0 ~/schedule]$
[observer@mpsr-srv0 ~/schedule]$ ~/automatic_mode.py schedule.sch
```

Limitations 🥺

- Required constant human supervision
- Any interruption would require a restart of the program with a modified schedule file
- Not suited for long dedicated surveys with large number of pointings

# SMIRF (Survey for Magnetars, Intermittent pulsars, RRATs and FRBs)

Fully autonomous software

Dynamic scheduling based on a static database of sources, tobs and cadence

Optimises schedule to reduce the amount of slew required

Removed the need for constant supervision

For a large survey - observer's role reduced to hitting Start/Stop button, once the source db was set-up.

Limitations 🥺

- Does not allow for custom sources to be observed at request
- Could not observe calibration sources
- For a fully autonomous system, it lacked a tracking mechanism for system health

# Other things to keep an eye on

Disk space (we record ~2 TB of data every day) - process candidates and delete on a daily basis

Packet drops happening due to bottlenecks in the data processing pipeline

Observing proximity to the sun

Temperature and load on servers

…

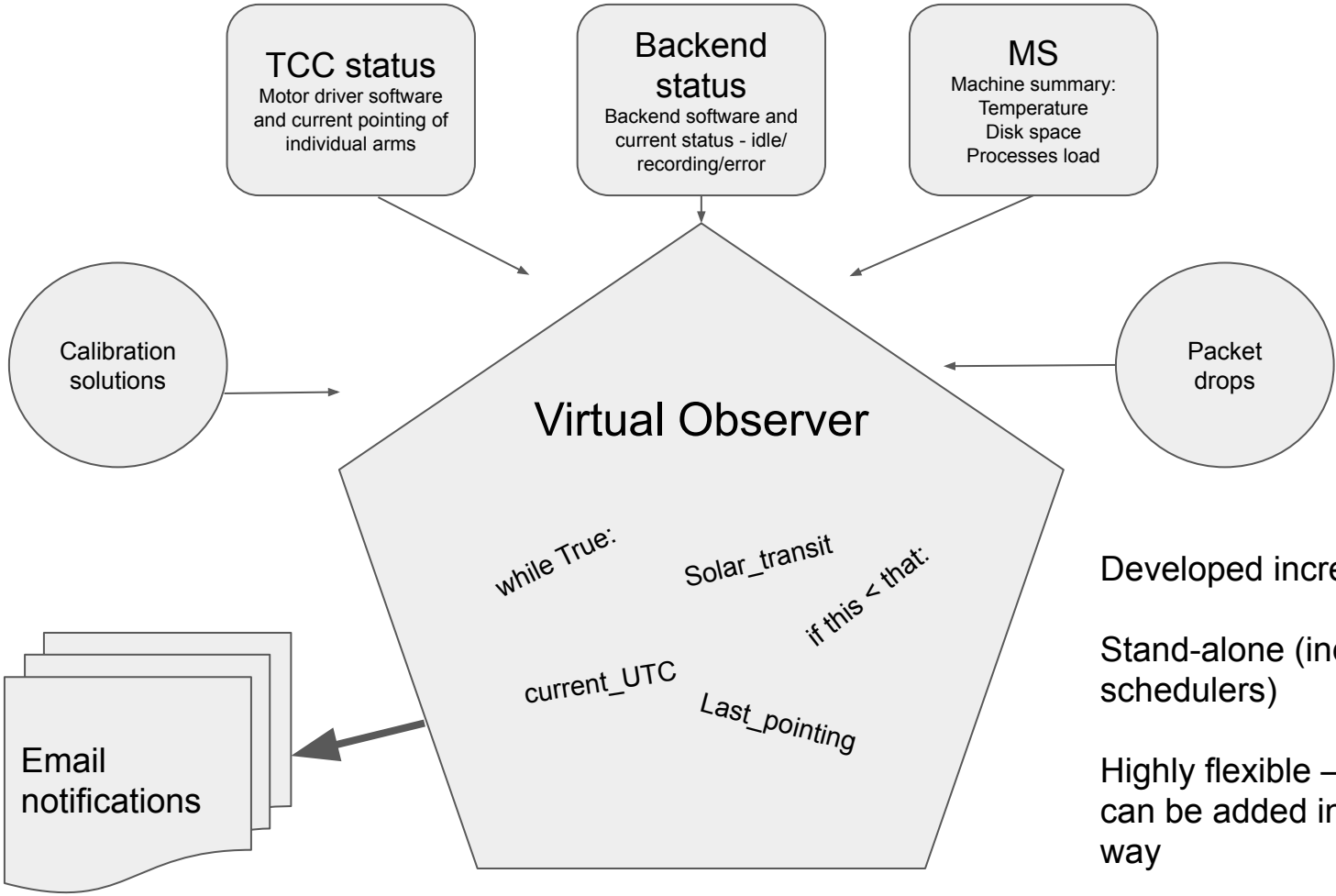
# Sad life of a PhD student

Chris and myself have been the de-facto Observer In Charge for ~2+ years.

Cannot keep an eye 24x7. Required reminders to observe sources. Prone to human error (forgetting things).

Telescope hardware is old - frequent mechanical failures (broken arm) resulting in abrupt termination of schedulers. If it goes unnoticed, many hours wasted.

*Simple tasks, but become cumbersome when having to do 24x7 for 2+ years.*



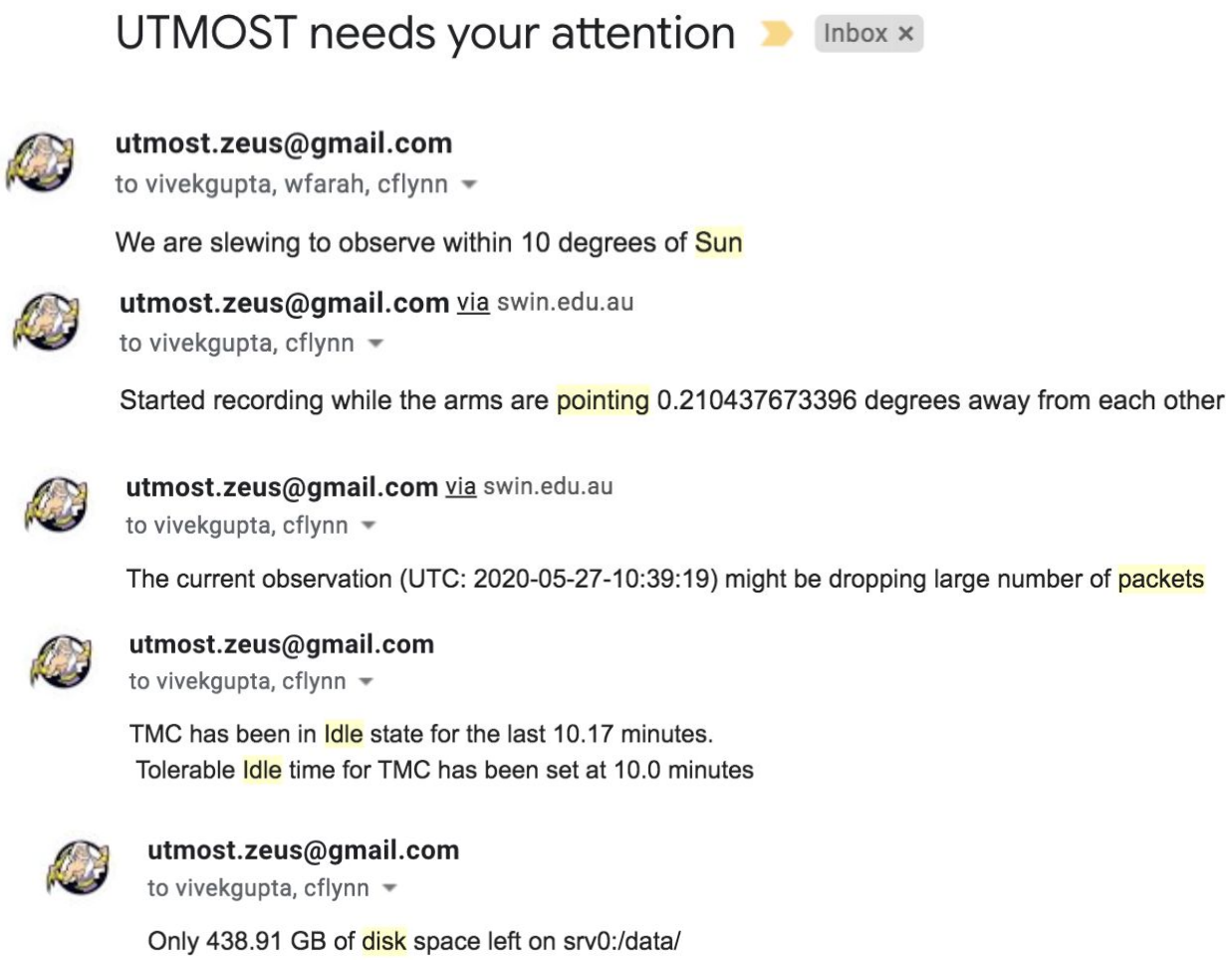

Developed incrementally

Stand-alone (indep. of schedulers)

Highly flexible – features can be added in a modular way

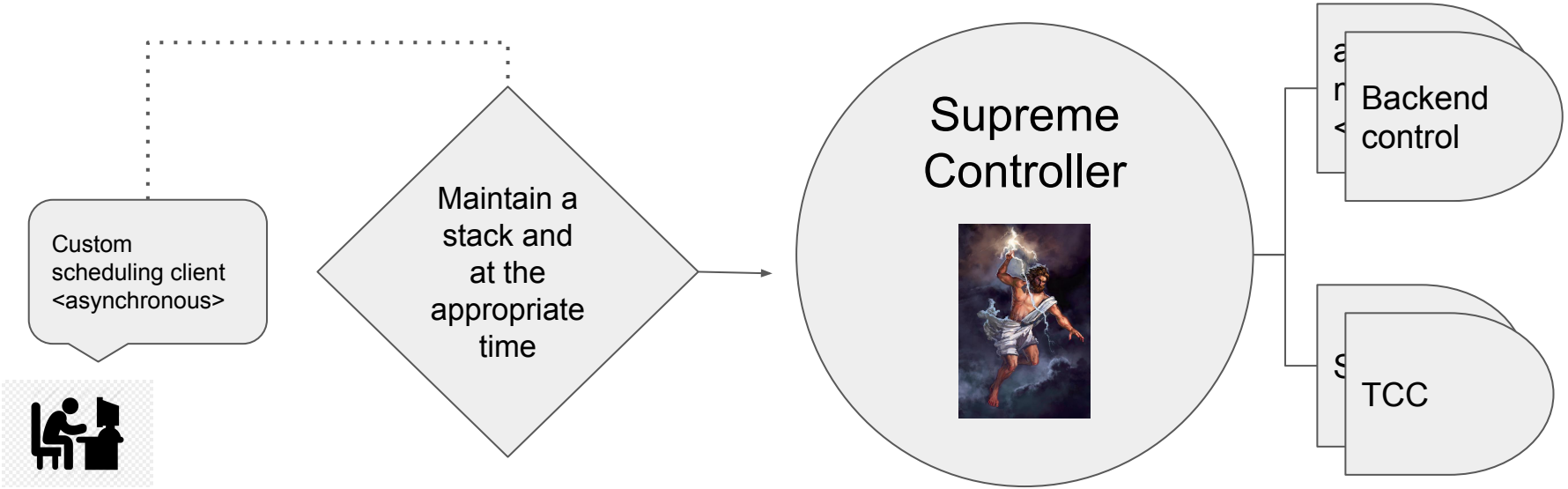
# Emails

- Made life much easier
- Caught rare exceptions - fixing bugs in SMIRF and automatic_mode



# Supreme Controller

After commissioning the Virtual Observer, started to develop an controller that could toggle SMIRF ON/OFF and take requests for custom sources to be observed in the middle of a SMIRF survey.

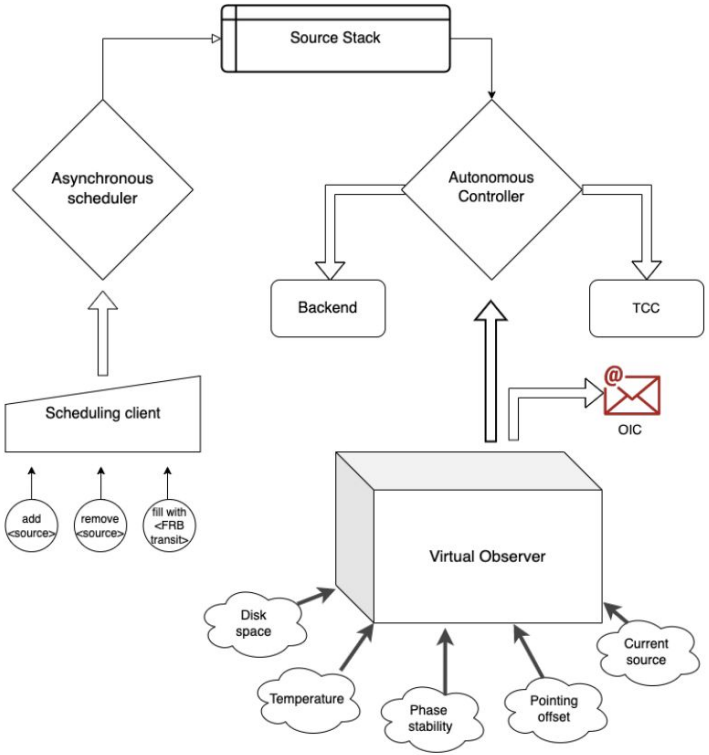


# Dynamic scheduler

```
[nsscheduler@mpsr-srv0 Hades]$ hoc
Attempting to connect to hos on ('localhost', 11310)
ready

>> show
Scheduler status: Active, FillWith = IDLE, Current Source = None
LST now: 4:33:20, Next source starts in -1.0 minutes
SourceStack = []

>>
>> add J1644-4559
OK

>>
>> show
Scheduler status: Active, FillWith = IDLE, Current Source = None
LST now: 4:33:44, Next source starts in 730.6 minutes
SourceStack = [J1644-4559]
```

```
>> help
  Try one of these:
  -show
  -fill_with [Filler Name] (for e.g., say 'idle' to keep the telescope idle between sources) | E.g. fill_with IDLE | fill_with FRBTRANSIT_<DEC>_<save_fb>
  -start
  -add [source name],<tobs (sec)>,<mode (xcorr/raw/tb/fb/fbtb/fb_transit)>,<save_fb (True/False)>,<use_outrigger (True/False)> | E.g. add J1644-4559
  -remove [source name] | E.g. remove J1644-4559
  -clear | Clears the whole source stack instantly
  -describe [source name/all] | E.g. describe J1644-4559. Describe all will list all sources and their properties
  -dump (saves the current source stack to /home/observer/schedule/hades_source_stack.txt
  -load [file_name/-] | E.g. load /home/observer/schedule/hades_source_stack.txt. load - will load the last active schedule before the server was killed.
  -pause
  -sleep [sleep_time (min)] (pauses observing for <sleep_time> minutes) | E.g. sleep 10
  -continue
  -done
  -kill (should be used sparingly). The current scheduled will be saved in a temp file automatically and can be reloaded using 'load -'.
```

# Combining the VO to the controller




VO could now not just send emails to the OIC, but also connect to the AC over a socket and ask it to take corrective action:

- If drives fail, stop slewing and start indefinite FRB transit obs at the current location
- If temp is too high, pause observing for 15 mins and resume again briefly, bail out after 3 attempts

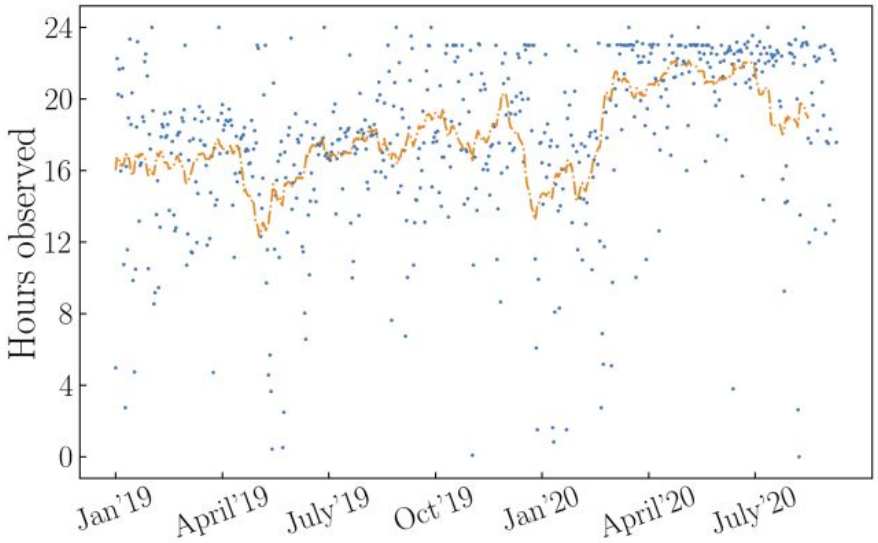
# Benefits

Resulted in significant improvement in the avg time-on-sky

Went from ~17 hours/day to ~22 hours/day (yielded ~10% more FRBs than would have got in the same duration at the initial efficiency)

Much fewer human interruptions

Happy PhD student – adapted to be used by the NS arm



# The trilogy

*The preserver*

Virtual Observer

*The creator*

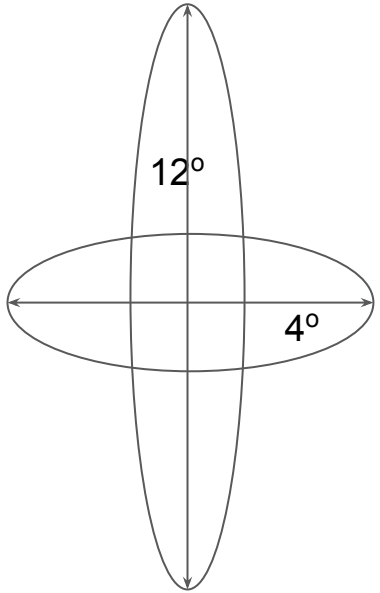
Dynamic scheduler

Autonomous Controller

*The destroyer*

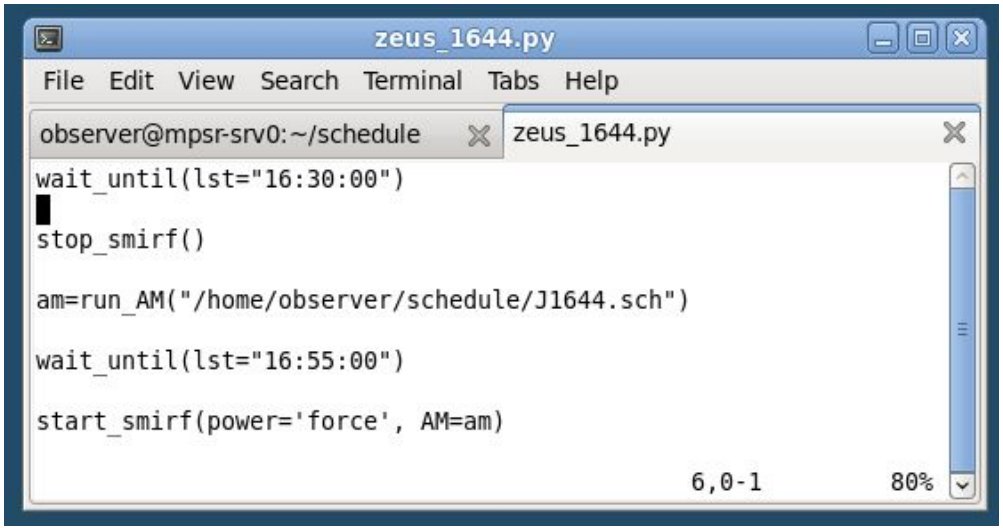
Questions?

—----------Backup slides—----------


12°
4°

# First steps

A script which is LST aware, and can toggle the existing schedulers. (Video)

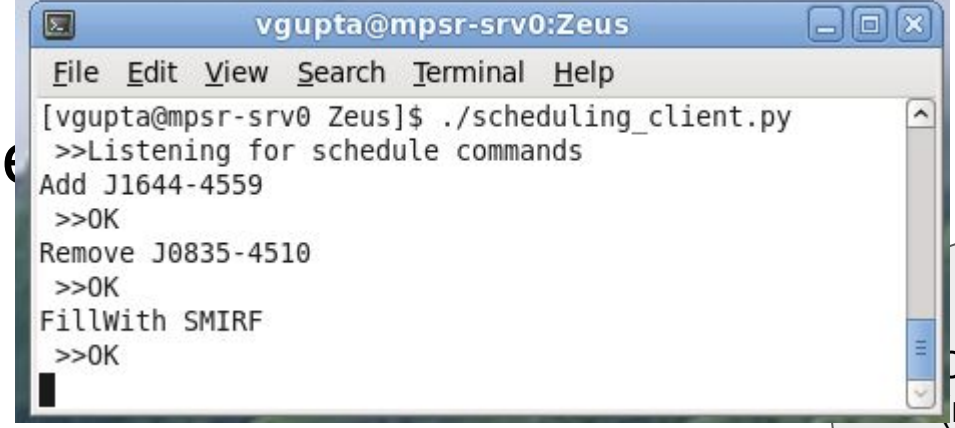



Once this was automated, we tended to forget about the telescope. Most problems went unnoticed. A need for a virtual_observer was obvious.

# Dynamic schedule

Highly asynchronous

Needs testing




Controller (real-time scheduling)

SourceStack (sorted with R.A.); FillGapsWith(SMIRF)

New sour-ce

Dyn. Sched. (sorts)

?

Virtual Observer