Field System Status

Dave Horsley[1], Ed Himwich[1], John Gipson[1], and Jonathan Quick[2]

[1]NVI/GSFC

[2]HartRAO

October 2018 EVN TOG

FS Linux Distribution

FSL9 — Current standard

- Based on Debian 7 “Wheezy”
- Most stations using it (hopefully not too many on Lenny)
- LTS ended June 2018, but still getting some occassional security updates

FSL10 — Next standard

- In development
- Will be based on Debian 9 “Stretch”
- Dual 32/64bit support (x86/x86-64 a.k.a. “amd64”)
- IT security requirements may affect the distribution
  - maybe optional image which conforms to NASA IT security
  - still discussion here...
- Expected end of 2018/early 2019

Current FS Release — v9.13.0

Released 2018-09-28.
VGOS support still in seperate v9.12 branch.

Server/Client mode

- Separate the FS into daemon server and UI client
- Replicated FS output to multiple clients (not just log output)
- supports autoftp and fs.prompt
- API could be implemented by other client (eg. eRc)
- Reference client reproduces X11 UI. A text-only client in development for high latency environments
- Opt-in for v9.13.0, default in future

Current FS Release — v9.13.0

Drudg

- Ad hoc support in `drudg` for `sched` “stagger start mode”

DBBC2 DDC

- Two VSI output modes with a FiLa10G and Ethernet recorder
- Support for `geo2` mode with firmware vers $\geq$ 106 for single VSI output on VSI1 output.
- Support for single VSI output on VSI2 for modes `wastro` and `geo2`

Also:

- Improvements to the `mk5c_mode` command to fully support Mark 5C and FlexBuff recording for both 5B/Ethernet and VDIF recording with `jive5ab`.

Current FS VGOS Release — v9.12.11

v9.12.11 Released 2017-03
No changes released

Current Monitoring System

This is a supplementary suite developed at GSFC.
Previously had a few other names. Now called MAS at NASA.
VLBI Specific plugins are implemented as Telegraf plugins with source provided in FS repo.
Current support:

- FS
  - tracking, data-valid, log, ...
- Met server
- RDBE tsys, pcal phase and amp (VGOS)
- MODBUS antennas

We feed these into InfluxDB and Grafana.

FS Future Release

- Phase-cal extraction and band pass plots.
  - Several stations have implemented this in various ways. We'd like to collect experiences and develop a standard interface and maybe implementation in the FS.
- 64bit support
- VEX2 support
- DBBC PFB Continuous Cal support
- Cal control improvements (more later. . .)
- Move FS to version control (Git)
- Add support for "scanning on the fly" for pointing checks
- Ethernet/serial converter support
- Ethernet/GPIB converter support

FS Future Release — Server/client

- Additional clients (terminal, browser, eRc?)
- Standardise API
- Add support VGOS branch
- Security model? Currently SSH, but something more fine-grained my be useful but may be too much work. Or maybe leave it up to eRc.

FS Future Release — VGOS branch (9.12)

- Initial Support for DBBC3 for VGOS observations
  - Inital support for Tsys and SEFD with DBBC3 backends.
  - Will provide a test case to work out any communication issues that may exist.
- Merge FS VGOS branch into main FS branch
- Add support for RDBE2
- Add full support for DBBC3

FS Future Release — Longer Term

Better support for high level languages

- Go
  - I have a Go library that can access the shared memory and a tool built on it to query it using a C-like syntax.
  - In general, hard to use as the field names are too opaque
  - Preventing data races is subtle (if not impossible with existing code)
- Python
  - Already some offline FS tools in Python 2, need to be upgraded to Python 3
  - Python 3 in Wheezy is too old to be useful, Python 2 support ends in a year (`https://pythonclock.org/`)
  - OS level Python support should get better once we move to Debian 9
  - Could use `cffi` to interact with FS, but would suffer the same general problems as with Go
- Probably need an API rather than shared memory

It would be very helpful to have:

- Feedback on bugs that are occurring in the field
- Input on what features are still needed
- Any other requests . . .

Acknowledgements

This presentation has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562 [RadioNet]

Proposed cal control improvements

- Add a line to RXG file for cal method
  - Allows specification per receiver
  - Options: continuous, on/off, hot/cold, none
  - Requires changes to gnplt and antabfs.py
- Use `if` command to select method in caltsys
- Chopper wheel and hot/cold load support
  - `caltsys` will take longer
  - Schedulers should consider allowing more PREOB time
  - Station provides a local program to calculate Tcal
  - Command line input: LO freq., pol., center frequency, met. temp, ...
  - Output: Tcal value
- What if different bands have different on/off cal control methods?
- Stations should implement local control in antcon
- It might be possible to pass LO freq. information