



Global VLBI Interfaces

JUMPING JIVE WP8



European VLBI Network



European VLBI Network

Configuration



European VLBI Network

Configuration

Verification



European VLBI Network

Configuration

Verification

VLBI - quite literally - only works if:

- all stations are:
 - observing the same position on the sky
 - at the same observing wavelength
- all equipment operates:
 - according to requested configuration
 - synchronized to sub-microsecond accuracy



European VLBI Network

Configuration

Task 8.1

Re-factoring of legacy scheduling software

Verification

Task 8.2

Centralized monitoring system



Global VLBI Network



Mark IV Correlator



MarkIV recording terminal



VLBA4 recording terminal



Global VLBI Network

Configuration

Task 8.1

Legacy scheduling software SCHED

- Developed at NRAO “*since the late 1970’s for Mark II VLBI scheduling.*”
- Programming language FORTRAN77
 - compiled code
 - only available option
- Not at all suited for
 - flexibility
 - user input processing
- Depends on another very old tool PGPLOT
 - FORTRAN graphics/plotting library, almost uncompileable these days



[HD] Steve Jobs - iPhone Introduction in 2007 (Complete)

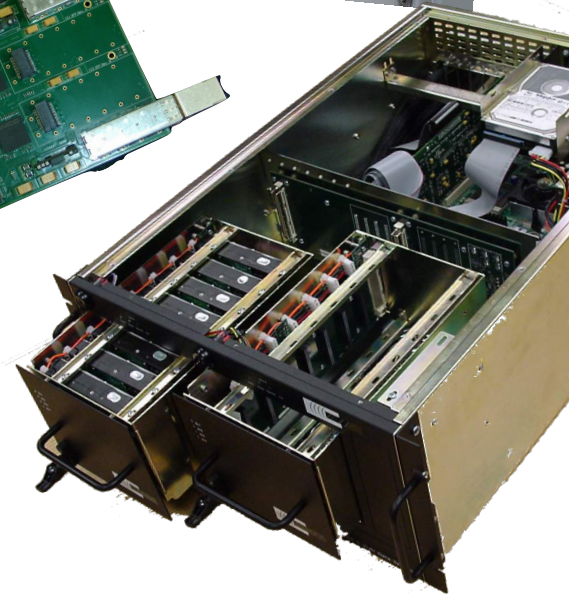
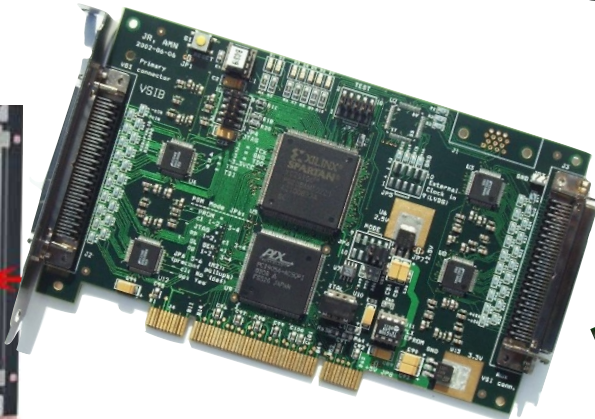
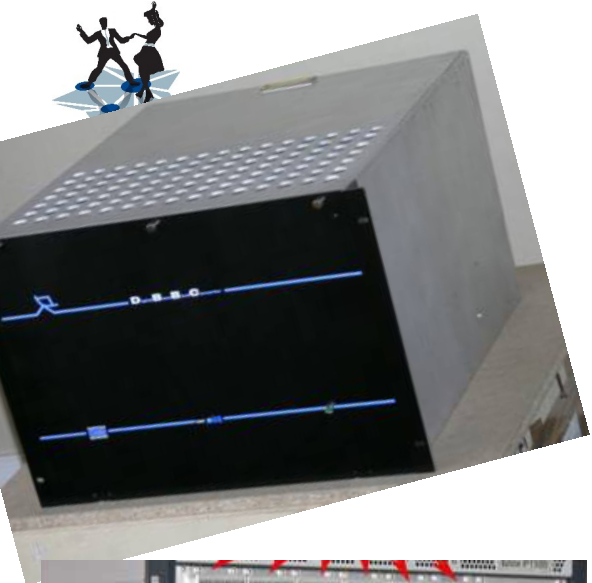


Revolutionary mobile phone



▶ ⏪ 🔊 1:57 / 1:19:09







Global VLBI Network

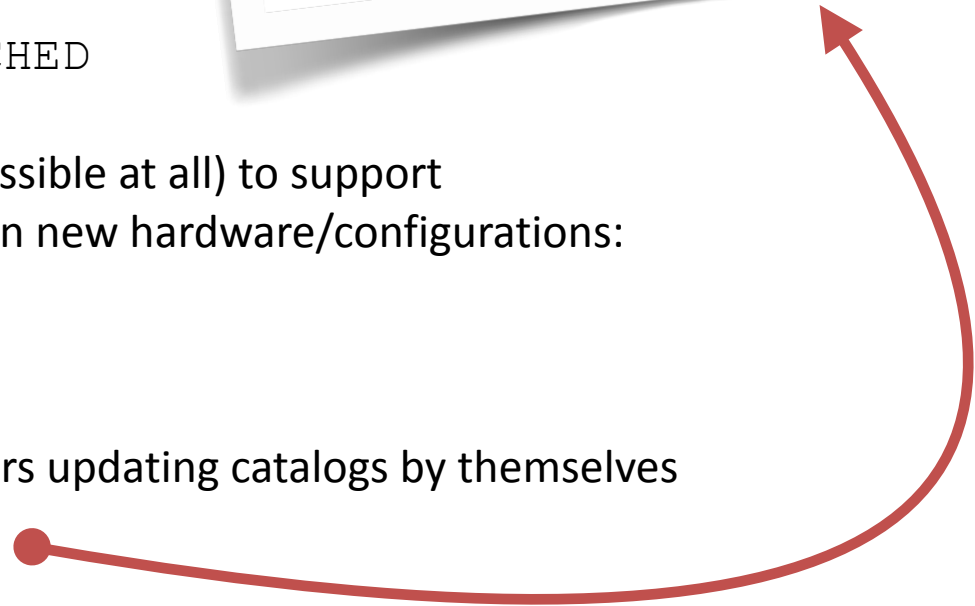
Configuration

Task 8.1

Legacy scheduling software SCHED

- No official support anymore
- New hardware difficult (if possible at all) to support
- Distribution of information on new hardware/configurations:
 - difficult
 - slow
 - manual
 - depends critically on users updating catalogs by themselves

Scheduling EVN experiments extremely hard,
time consuming and error prone





Global VLBI Network

Configuration

Task 8.1

Current:

- Python* is *the* scientific programming language within astronomy
 - Interpreted
 - Dynamic, flexible, oriented towards input parsing
 - More than 60 000 external packages which add functionality
- New (young) users get taught Python
- Modern websites / portals set high expectation level of (new, young) users

[*] <https://www.python.org/>



Global VLBI Network

Configuration

Task 8.1

Current:

- Python* is *the* scientific programming language of choice by
 - Interpreted
 - Dynamic typing
 - More than 1000 third party modules add functionality
- New (young) users get taught Python
- Modern websites / portals set high expectation level of (new, young) users

FORTRAN ⇒ Python

[*] <https://www.python.org/>



Global VLBI Network

Configuration

Task 8.1

Approach:

- do NOT rewrite all of SCHED (80 000 lines-of-code)
- it has > 30 years of domain expertise aggregated

Instead: make SCHED Python extension modules

- main loop is now in Python, not FORTRAN
- then re-factor problematic parts
- replace inflexible FORTRAN parts by flexible Python parts
- extend functionality by exploiting external Python packages
- address maintainability issues



Global VLBI Network

Configuration

Task 8.1

Approach:

- do NOT rewrite all of SCHED (80 000 lines-of-code)
- it has > 30 years of domain expertise aggregated

Instead: make SCHED Python extension modules

- main loop is now in Python, not FORTRAN
- then re-factor problematic parts
- replace inflexible FORTRAN parts by flexible Python parts
- extend functionality by exploiting external Python packages
- address maintainability issues



Global VLBI Network

Configuration

Task 8.1

Approach:

- do NOT rewrite all of SCHED (80 000 lines-of-code)
- it has > 30 years of domain expertise aggregated

Instead: make SCHED Python extension modules

- main loop is now in Python, not FORTRAN
- then re-factor problematic parts
- replace inflexible FORTRAN parts by flexible Python parts
- extend functionality by exploiting external Python packages
- address maintainability issues



Global VLBI Network

Configuration

Task 8.1

Approach:

- do NOT rewrite all of SCHED (80 000 lines-of-code)
- it has > 30 years of domain expertise aggregated

Instead: make SCHED Python extension modules

- main loop is now in Python, not FORTRAN
- then re-factor problematic parts
- replace inflexible FORTRAN parts by flexible Python parts
- extend functionality by exploiting external Python packages
- address maintainability issues



Global VLBI Network

Configuration

Task 8.1

Approach:

- do NOT rewrite all of SCHED (80 000 lines-of-code)
- it has > 30 years of domain expertise aggregated

Instead: make SCHED Python extension modules

- main loop is now in Python, not FORTRAN
- then re-factor problematic parts
- replace inflexible FORTRAN parts by flexible Python parts
- extend functionality by exploiting external Python packages
- address maintainability issues



Global VLBI Network

Configuration

Task 8.1

JIVE staff effort has led to `pySCHEd`* (see deliverable D8.3 for details)

- ~40 000 lines of code Pythonized (50 %)
- distributed as Python module
 - deployment extremely simple: `pip install pySCHEd`
- source code available through collaborative `github` platform
 - issue reporting for users, documentation
- catalogs centrally maintained on `github` too
- `sched` program is now a shell around the actual program
 - checks for updates on `github` and informs user about new version
 - automatically updates catalogs at startup

[*] <https://github.com/jive-vlbi/sched/>



Global VLBI Network

Configuration

Task 8.1

JIVE staff effort has led to `pySCHEd`* (see deliverable D8.3 for details)

- ~40 000 lines of code Pythonized (50 %)
- distributed as Python module
 - deployment extremely simple: `pip install pySCHEd`
- source code available through collaborative `github` platform
 - issue reporting for users, documentation
- catalogs centrally maintained on `github` too
- `sched` program is now a shell around the actual program
 - checks for updates on `github` and informs user about new version
 - automatically updates catalogs at startup

[*] <https://github.com/jive-vlbi/sched/>



Global VLBI Network

Configuration

Task 8.1

JIVE staff effort has led to `pySCHEd`* (see deliverable D8.3 for details)

- 20 000 lines of code Pythonized (xx %)
- distributed as Python module
 - deployment extremely simple: `pip install pythonSCHEd`
- source code available through collaborative `github` platform
 - issue reporting for users, documentation
- catalogs centrally maintained on `github` too
- `sched` program is now a shell around the actual program
 - checks for updates on `github` and informs user about new version
 - automatically updates catalogs at startup

[*] <https://github.com/jive-vlbi/sched/>



Global VLBI Network

Configuration

Task 8.1

JIVE staff effort has led to `pySCHEd`* (see deliverable D8.3 for details)

- 20 000 lines of code Pythonized (xx %)
- distributed as Python module
 - deployment extremely simple: `pip install pythonSCHEd`
- source code available through collaborative `github` platform
 - issue reporting for users, documentation
- catalogs centrally maintained on `github` too
- `sched` program is now a shell around the actual program
 - checks for updates on `github` and informs user about new version
 - automatically updates catalogs at startup

[*] <https://github.com/jive-vlbi/sched/>



Global VLBI Network

Configuration

Task 8.1

JIVE staff effort has led to `pySCHEd`* (see deliverable D8.3 for details)

- 20 000 lines of code Pythonized (xx %)
- distributed as Python module
 - deployment extremely simple: `pip install pythonSCHEd`
- source code available through collaborative `github` platform
 - issue reporting for users, documentation
- catalogs centrally maintained on `github` too
- `sched` program is now a shell around the actual program
 - checks for updates on `github` and informs user about new version
 - automatically updates catalogs at startup

[*] <https://github.com/jive-vlbi/sched/>



Global VLBI Network

Configuration

Task 8.1

Objective:

“The end product will be a modernised version of SCHED that will be far easier to adapt, written in a widely used and well-known programming language. It will be usable for all aspects of the proposal-observational cycle, which means during the proposal phase, the programming by the PI and finally the actual generation of an observing schedule by the JIVE staff.”



First-time user of EVN

Nicolaus Copernicus University
Toruń, Poland

Things that I liked about pySCHED:

- easy to deploy. All I have to do is just type „pip install pythonSCHED”. I never managed to install original SCHED on my computer and thus, I was forced to work remotely on computer with installed SCHED via SSH.
- no need to copy the frequency setup information, sent by JIVE – pySCHED takes latest frequency setup
- checks for updates
- plotting via matplotlib – it’s easy to zoom plots to examine results closer, also provides possibility to save plot as png/pdf/eps etc, making it easier to share with coworkers.
- normal-looking self-explanatory UI (as far as I know, it is based on QT)
- open source
- actively maintained (latest version 1.19 is from 2021-07-09)
- github page (<https://github.com/jive-vlbi/sched>)

To be honest, I've never encountered a bug, when using pySCHED.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



First-time user of EVN

Nicolaus Copernicus University
Toruń, Poland

Things that I liked about pySCHED:

- **easy to deploy.** All I have to do is just type „pip install pythonSCHED”. I never managed to install original SCHED on my computer and thus, I was forced to work remotely on computer with installed SCHED via SSH.
- **no need to copy the frequency setup** information, sent by JIVE – pySCHED takes latest frequency setup
- **checks for updates**
- plotting via matplotlib – it’s easy to zoom plots to examine results closer, also provides possibility to save plot as png/pdf/eps etc, making it easier to share with coworkers.
- normal-looking self-explanatory UI (as far as I know, it is based on QT)
- open source
- **actively maintained** (latest version 1.19 is from 2021-07-09)
- github page (<https://github.com/jive-vlbi/sched>)

To be honest, I've never encountered a bug, when using pySCHED.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Expert in SCHED + EVN, (V)LBA

pySCHED user review -

- I first had to start using SCHED 4 years ago but I could never successfully install it on my Mac OS or Linux systems.
- I was, thus, using SCHED by sshing into the ATCA server that had SCHED pre-installed.
- Checking plots and .sum files always took a long time for me and involved moving files from the ATCA server to my local machine.
- Getting introduced to pySCHED was a huge relief. I was able to install it on my Mac OS system and easily use it to edit and work on my .key files. The plots generated also had a lot of flexibility and was more intuitive to use.
- I will continue to use pySCHED here on out.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Expert in SCHED + EVN, (V)LBA

pySCHED user review -

- I first had to start using SCHED 4 years ago but I could never successfully install it on my Mac OS or Linux systems.
- I was, thus, using SCHED by sshing into the ATCA server that had SCHED pre-installed.
- Checking plots and .sum files always took a long time for me and involved moving files from the ATCA server to my local machine.
- Getting introduced to pySCHED was a huge relief. I was able to install it on my Mac OS system and easily use it to edit and work on my .key files. The plots generated also had a lot of flexibility and was more intuitive to use.
- I will continue to use pySCHED here on out.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Expert in SCHED + EVN, (V)LBA

pySCHED user review -

- I first had to start using SCHED 4 years ago but I could never successfully install it on my Mac OS or Linux systems.
- I was, thus, using SCHED by sshing into the ATCA server that had SCHED pre-installed.
- Checking plots and .sum files always took a long time for me and involved moving files from the ATCA server to my local machine.
- Getting introduced to **pySCHED** was a huge relief. I was able to install it on my Mac OS system and easily use it to edit and work on my .key files. The plots generated also had a lot of flexibility and was more intuitive to use.
- I will continue to use pySCHED here on out.

Note: **pySCHED** has been available to the public since Sept 2019 (2 yrs ago)



Many years expert in SCHED

I had some difficulty with PGPLOT in my new computer so I had to make the move to python

What I liked about pySCHED was that I could prepare my keyfile before the setinis were sent around by jobs because the station, loc, freq, catalogs etc. are all pulled/updated when running pySCHED.

I also liked the plotting.

pySCHED is familiar enough in its look and operation with comparison to SCHED.

I imagine it will be easier to add new functionality in pySCHED which will be helpful, such as highlighting station scans where the tsys was not done on-source, this SCHED warning usually meant having to dig into sumfiles which took time because SCHED never told you when those not-good-tsys scans occur in the schedule.

I like not having to take responsibility of maintaining catalog files locally. Any source catalogs for targets, or user defined frequency settings, can be done inside the keyfile. So I like that I have control when I want it and not having to worry when I dont want to.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Many years expert in SCHED

I had some difficulty with PGPLOT in my new computer so I had to make the move to python

What I liked about pySCHED was that I could prepare my keyfile before the setinis were sent around by jobs because the station, loc, freq, catalogs etc. are all pulled/updated when running pySCHED.

I also liked the plotting.

pySCHED is familiar enough in its look and operation with comparison to SCHED.

I imagine it will be easier to add new functionality in pySCHED which will be helpful, such as highlighting station scans where the tsys was not done on-source, this SCHED warning usually meant having to dig into sumfiles which took time because SCHED never told you when those not-good-tsys scans occur in the schedule.

I like not having to take responsibility of maintaining catalog files locally. Any source catalogs for targets, or user defined frequency settings, can be done inside the keyfile. So I like that I have control when I want it and not having to worry when I dont want to.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Many years expert in SCHED

I had some difficulty with PGPLOT in my new computer so I had to make the move to python

What I liked about pySCHED was that I could prepare my keyfile before the setinis were sent around by jobs because the station, loc, freq, catalogs etc. are all pulled/updated when running pySCHED

I also liked the plotting.

pySCHED is familiar enough in its look and operation with comparison to SCHED.

I imagine it will be easier to add new functionality in pySCHED which will be helpful, such as highlighting station scans where the tsys was not done on-source, this SCHED warning usually meant having to dig into sumfiles which took time because SCHED never told you when those not-good-tsys scans occur in the schedule.

I like not having to take responsibility of maintaining catalog files locally. Any source catalogs for targets, or user defined frequency settings, can be done inside the keyfile. So I like that I have control when I want it and not having to worry when I don't want to.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Many years expert in SCHED

I had some difficulty with PGPLOT in my new computer so I had to make the move to python

What I liked about pySCHED was that I could prepare my keyfile before the setinis were sent around by jobs because the station, loc, freq, catalogs etc. are all pulled/updated when running pySCHED.

I also liked the plotting.

pySCHED is familiar enough in its look and operation with comparison to SCHED.

I imagine it will be easier to add new functionality in pySCHED which will be helpful, such as highlighting station scans where the tsys was not done on-source, this SCHED warning usually meant having to dig into sumfiles which took time because SCHED never told you when those not-good-tsys scans occur in the schedule.

I like not having to take responsibility of maintaining catalog files locally. Any source catalogs for targets, or user defined frequency settings, can be done inside the keyfile. So I like that I have control when I want it and not having to worry when I dont want to.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Many years expert in SCHED

I had some difficulty with PGPLOT in my new computer so I had to make the move to python

What I liked about pySCHED was that I could prepare my keyfile before the setinis were sent around by jobs because the station, loc, freq, catalogs etc. are all pulled/updated when running pySCHED.

I also liked the plotting.

pySCHED is familiar enough in its look and operation with comparison to SCHED.

I imagine it will be easier to add new functionality in pySCHED which will be helpful, such as highlighting station scans where the tsys was not done on-source, this SCHED warning usually meant having to dig into sumfiles which took time because SCHED never told you when those not-good-tsys scans occur in the schedule.

I like not having to take responsibility of maintaining catalog files locally. Any source catalogs for targets, or user defined frequency settings, can be done inside the keyfile. So I like that I have control when I want it and not having to worry when I dont want to.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Fast response/new mode

----- Forwarded message -----

Date: **Mon, 5 Oct 2020 13:51:27**

Subject: Re: Fwd: EY035 A and B key files

I just uploaded a new version of pySCHED which will allow the setup for EY035B.

XXX XXXXX <xxx@jive.eu> **10/05/20 11:53 AM >>>**

Hi Xxx --

Trying out-stations at 1-bit sampling in order to cover the full 128MHz range is something YYY wants for his L-band obs. e-MERLIN people say they can do that, but would need to use 128MHz filters to keep within the single WIDAR-board (thus one "output" per station). So in this case, there would need to be an allowed 128MHz filter width and a corresponding 256Msamp/s sampling rate (each checked in different sub-routines in regular sched (samprate in chkdbbc.f, filter-BW in chkdbfq.f).

Whether the 128MHz filter & 256Msamp/s is only limited to 1-bit sampling in the long term, I don't know. For now, it seems your algorithm should work.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Fast response/new mode

----- Forwarded message -----

Date: **Mon, 5 Oct 2020 13:51:27**

Subject: Re: Fwd: EY035 A and B key files

I just uploaded a new version of pySCHED which will allow the setup for EY035B.

XXX XXXXX <xxx@jive.eu> **10/05/20 11:53 AM** >>>

Hi Xxx --

Trying out-stations at 1-bit sampling in order to cover the full 128MHz range is something YYY wants for his L-band obs. e-MERLIN people say they can do that, but would need to use 128MHz filters to keep within the single WIDAR-board (thus one "output" per station). So in this case, there would need to be an allowed 128MHz filter width and a corresponding 256Msamp/s sampling rate (each checked in different sub-routines in regular sched (samprate in chkdbbc.f, filter-BW in chkdbfq.f).

Whether the 128MHz filter & 256Msamp/s is only limited to 1-bit sampling in the long term, I don't know. For now, it seems your algorithm should work.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Fast response/new mode

----- Forwarded message -----

Date: Mon, 5 Oct 2020 13:51:27

Subject: Re: Fwd: EY035 A and B key files

I just uploaded a new version of pySCHED which will allow the setup for EY035B.

XXX XXXXX <xxx@jive.eu> 10/05/20 11:53 AM >>>

Hi Xxx --

Trying out-stations at 1-bit sampling in order to cover the full 128MHz range is something YYY wants for his L-band obs. e-MERLIN people say they can do that, but would need to use 128MHz filters to keep within the single WIDAR-board (thus one "output" per station). So in this case, there would need to be an allowed 128MHz filter width and a corresponding 256Msamp/s sampling rate (each checked in different sub-routines in regular sched (samprate in chkdbbc.f, filter-BW in chkdbfq.f).

Whether the 128MHz filter & 256Msamp/s is only limited to 1-bit sampling in the long term, I don't know. For now, it seems your algorithm should work.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Fast response/fixing problems

----- Forwarded message -----

Date: Fri, 27 Sep 2019 14:46:44

Subject: RE: AB1234 on the Oct/Nov EVN block schedule

Hi Yyy,

This totally worked! Many thanks!!!

Regards.

Xxx.

Sent: Friday, September 27, 2019 1:34 PM

Subject: Re: AB1234 on the Oct/Nov EVN block schedule

Hi Xxx,

pySCHED wasn't based on the latest version of SCHED. I have just uploaded version 1.2.0 of pySCHED to pypi, this version is based on the FORTRAN code which can handle the 4th Nyquist zone. To update your version of pySCHED, the following command should do the trick:

```
pip install --upgrade pythonSCHED
```

The new version should be able to process your key file properly.

Note: `pySCHED` has been available to the public since Sept 2019 (2 yrs ago)



Fast response/fixing problems

----- Forwarded message -----

Date: Fri, 27 Sep 2019 14:46:44

Subject: RE: AB1234 on the Oct/Nov EVN block schedule

Hi Yyy,

This totally worked! Many thanks!!!

Regards.

Xxx.

Sent: Friday, September 27, 2019 1:34 PM

Subject: Re: AB1234 on the Oct/Nov EVN block schedule

Hi Xxx,

pySCHED wasn't based on the latest version of SCHED. I have just uploaded version 1.2.0 of pySCHED to pypi, this version is based on the FORTRAN code which can handle the 4th Nyquist zone. To update your version of pySCHED, the following command should do the trick:

```
pip install --upgrade pythonSCHED
```

The new version should be able to process your key file properly.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Fast response/fixing problems

----- Forwarded message -----

Date: Fri, 27 Sep 2019 14:46:44

Subject: RE: AB1234 on the Oct/Nov EVN block schedule

Hi Yyy,

This totally worked! Many thanks!!!

Regards.

Xxx.

Sent: Friday, September 27, 2019 1:34 PM

Subject: Re: AB1234 on the Oct/Nov EVN block schedule

Hi Xxx,

pySCHED wasn't based on the latest version of SCHED. I have just uploaded version 1.2.0 of pySCHED to pypi, this version is based on the FORTRAN code which can handle the 4th Nyquist zone. To update your version of pySCHED, the following command should do the trick:

```
pip install --upgrade pythonSCHED
```

The new version should be able to process your key file properly.

Note: pySCHED has been available to the public since Sept 2019 (2 yrs ago)



Installation issues

Following the GIT README, I was able to successfully install the pySCHED via conda (I've created a SCHEDenv with your yaml). Once I did that, I tried to run a toy key example (in attach) provided somewhere in your archival - it has been a long time I didn't run sched and my key file it is a bit harder. Unfortunately, it seems it is not working. Here the output

```
(pySCHED) xxx@503d2c18ac40:/host/sched/TEST$ sched.py -p -k test.key
```

```
Welcome to pySCHED version 1.19.0 based on  
SCHED version: 11.6 Release 11.6; Feburary 2020
```

Bug reports, feature requests and other discussions can be posted on the GitHub page:

<https://github.com/jive-vlbi/sched>

pySCHED will use the catalogs under /home/xxx/.pysched unless \$SCHED is set.

Updating catalogs in /home/xxxx/.pysched

Catalogs in /home/xxx/.pysched are up-to-date.

Schedule version is missing.

No PINAME given.

No address specified.

No PI phone number specified.

No email address or fax number specified.

GETCOV: Cover information incomplete or missing.

SCHED: Processing input.

CHKSCN: 139 scans had more than half the antennas arrive on-source after the start time.

This could be normal if using duration scheduling with small gaps.

SUMOPE: Writing summary file uvcov.sum

Error running pySCHED: Cannot load backend 'Qt5Agg' which requires the 'qt5' interactive framework, as 'headless' is currently running



Installation issues

Following the GIT README, I was able to successfully install the pySCHED via conda (I've created a SCHEDenv with your yamll). Once I did that, I tried to run a toy key example (in attach) provided somewhere in your archival - it has been a long time I didn't run sched and my key file it is a bit harder. **Unfortunately, it seems it is not working.** Here the output

```
(pySCHED) xxx@503d2c18ac40:/host/sched/TEST$ sched.py -p -k test.key
```

```
Welcome to pySCHED version 1.19.0 based on  
SCHED version: 11.6 Release 11.6; Feburary 2020
```

*Bug reports, feature requests and other discussions can be posted on the GitHub page:
<https://github.com/jive-vlbi/sched>
pySCHED will use the catalogs under /home/xxx/.pysched unless \$SCHED is set.*

```
Updating catalogs in /home/xxxx/.pysched  
Catalogs in /home/xxx/.pysched are up-to-date.
```

```
Schedule version is missing.
```

```
No PINAME given.
```

```
No address specified.
```

```
No PI phone number specified.
```

```
No email address or fax number specified.
```

```
GETCOV: Cover information incomplete or missing.
```

```
SCHED: Processing input.
```

```
CHKSCN: 139 scans had more than half the antennas arrive on-source after the start time.
```

```
This could be normal if using duration scheduling with small gaps.
```

```
SUMOPE: Writing summary file uvcov.sum
```

Error running pySCHED: Cannot load backend 'Qt5Agg' which requires the 'qt5' interactive framework, as 'headless' is currently running



Installation issues

Support e-mail address at JIVE:

pysched@jive.eu



Global VLBI Network

Configuration

Task 8.1

Dissemination:

- <https://evlbi.org/evn-scheduling>
- Prepared tutorial at EVN Symposium 2020
 - cancelled because of COVID-19
- EVN Symposium 2021:
 - held online only, limited time slots - no tutorials



European VLBI Network

Verification

Task 8.2

Centralized monitoring system

2021.161.12:00:55.65#tpicd#tpcont/bu,10246,9536,cl,10159,9460,cu,9955,9153,ib,13983.20
2021.161.12:01:00.66#tpicd#tpcont/1l,10473,10974,1u,9669,10003,2l,9919,10691,2u,9794,10417,3l,9952,10836
2021.161.12:01:00.66#tpicd#tpcont/3u,9724,10501,4l,10034,10771,4u,10000,10865,ia,13991.79
2021.161.12:01:00.66#tpicd#tpcont/9l,10386,9888,9u,10081,9686,al,9739,9102,au,10148,9400,bl,10167,9488
2021.161.12:01:00.66#tpicd#tpcont/bu,10208,9510,cl,10180,9472,cu,10000,9187,ib,14011.13
2021.161.12:01:05.68#tpicd#tpcont/1l,9836,10334,1u,10429,10794,2l,9936,10749,2u,9790,10444,3l,9966,10874
2021.161.12:01:05.68#tpicd#tpcont/3u,10350,11199,4l,10047,10831,4u,9968,10861,ia,14032.66
2021.161.12:01:05.68#tpicd#tpcont/9l,9794,9305,9u,10092,9693,al,10282,9613,au,10158,9405,bl,9497,8835
2021.161.12:01:05.68#tpicd#tpcont/bu,9686,9001,cl,10196,9483,cu,9912,9105,ib,14019.74
2021.161.12:01:10.68#tpicd#tpcont/1l,9872,10347,1u,9706,10038,2l,9966,10752,2u,9838,10472,3l,10015,10891
2021.161.12:01:10.68#tpicd#tpcont/3u,9767,10562,4l,10105,10852,4u,10041,10906,ia,14049.90
2021.161.12:01:10.68#tpicd#tpcont/9l,9355,8893,9u,9074,8709,al,8793,8214,au,9154,8471,bl,9229,8588
2021.161.12:01:10.68#tpicd#tpcont/bu,9216,8576,cl,9202,8557,cu,8992,8251,ib,12365.36
2021.161.12:01:15.69#tpicd#tpcont/1l,9966,10495,1u,10008,10383,2l,10022,10873,2u,9888,10574,3l,9576,10485
2021.161.12:01:15.69#tpicd#tpcont/3u,9869,10711,4l,10108,10920,4u,9631,10520,ia,12263.28
2021.161.12:01:15.69#tpicd#tpcont/9l,9901,9426,9u,9675,9296,al,10245,9591,au,9736,9028,bl,9847,9190
2021.161.12:01:15.69#tpicd#tpcont/bu,9769,9099,cl,9748,9088,cu,10267,9450,ib,12380.55
2021.161.12:01:20.71#tpicd#tpcont/1l,9986,10507,1u,10027,10400,2l,10051,10893,2u,9912,10592,3l,10233,11187
2021.161.12:01:20.71#tpicd#tpcont/3u,9897,10735,4l,10153,10964,4u,10273,11205,ia,12267.04
2021.161.12:01:20.71#tpicd#tpcont/9l,9911,9431,9u,10290,9910,al,9762,9132,au,10334,9583,bl,9857,9193
2021.161.12:01:20.71#tpicd#tpcont/bu,10285,9585,cl,10298,9597,cu,10204,9395,ib,12393.86
2021.161.12:01:25.71#tpicd#tpcont/1l,10052,10584,1u,10086,10465,2l,10127,10970,2u,9978,10656,3l,9669,10585
2021.161.12:01:25.71#tpicd#tpcont/3u,9969,10814,4l,9713,10484,4u,9679,10566,ia,12443.43
2021.161.12:01:25.71#tpicd#tpcont/9l,9908,9441,9u,9681,9307,al,10259,9597,au,9745,9036,bl,9856,9198
2021.161.12:01:25.71#tpicd#tpcont/bu,9775,9105,cl,9761,9102,cu,9513,8741,ib,12395.76
2021.161.12:01:30.72#tpicd#tpcont/1l,10091,10597,1u,10124,10472,2l,10170,10987,2u,10020,10673,3l,10366,11288
2021.161.12:01:30.72#tpicd#tpcont/3u,10020,10828,4l,9777,10517,4u,10380,11296,ia,12485.52
2021.161.12:01:30.72#tpicd#tpcont/9l,9916,9431,9u,10299,9910,al,9767,9133,au,10343,9580,bl,9867,9195
2021.161.12:01:30.72#tpicd#tpcont/bu,10294,9593,cl,10315,9602,cu,10121,9295,ib,12395.76
2021.161.12:01:35.73#tpicd#tpcont/1l,10080,10603,1u,10108,10474,2l,10166,11008,2u,10014,10679,3l,9719,10623



European VLBI Network

```
[ccs]Okay->wc -l *.log
  3133 n21c2bd.log
 10422 n21c2ef.log
   3249 n21c2hh.log
   4201 n21c2ir.log
   7270 n21c2jb.log
   3189 n21c2mc.log
   3654 n21c2nt.log
   4200 n21c2o8.log
   1889 n21c2sv.log
   3247 n21c2t6.log
   6054 n21c2tr.log
   2721 n21c2ur.log
   2827 n21c2wb.log
   8396 n21c2ys.log
   4171 n21c2zc.log
 68623 total
```

Verification

Task 8.2

Centralized monitoring system

~1200 pages of text



European VLBI Network

Verification

Task 8.2

Many large text files can be processed by computer

- totally human unfriendly
- cannot quickly find errors or equipment good/bad stat
- cannot relate problems between sites
- no trends visible across experiments



European VLBI Network

Verification

Task 8.2

Text files (manually) uploaded *after* the experiment

- ➔ reasonable chance problem cannot be fixed anymore
- ➔ lost data + science



European VLBI Network

Verification

Task 8.2

The Technische Universität München developed centralized monitoring system

- using industry standard basis software
 - time-series specialized database
 - large set of measurement points available “out of the box”
- continuation of previously EC-funded project in NEXPreS; currently available:
 - tools specific to VLBI stations
 - RFI Monitoring from spectra, mobile version of web pages, analysis of critical situations in the cryogenic Dewar, Grafana and possible earth maps, automated data module tracker for Mark5 modules, dynamic ZABBIX maps, monitoring of additional systems using ZABBIX agents
 - software to extract monitoring data from log files as they are written
 - dashboards/overviews of interest to different user groups
 - scripts to upload data to centralized monitoring system in real time



European VLBI Network

Verification

Task 8.2

The Technische Universität München developed centralized monitoring system

- using industry standard basis software
 - time-series specialized database
 - large set of measurement points available “out of the box”
- continuation of previously EC-funded project in NEXPREs; currently available:
 - **tools specific to VLBI stations**
 - RFI Monitoring from spectra, mobile version of web pages, analysis of critical situations in the cryogenic Dewar, Grafana and possible earth maps, automated data module tracker for Mark5 modules, dynamic ZABBIX maps, monitoring of additional systems using ZABBIX agents
 - software to extract monitoring data from log files as they are written
 - dashboards/overviews of interest to different user groups
 - scripts to upload data to centralized monitoring system in real time



European VLBI Network

Verification

Task 8.2

The Technische Universität München developed centralized monitoring system

- using industry standard basis software
 - time-series specialized database
 - large set of measurement points available “out of the box”
- continuation of previously EC-funded project in NEXPreS; currently available:
 - tools specific to VLBI stations
 - RFI Monitoring from spectra, mobile version of web pages, analysis of critical situations in the cryogenic Dewar, Grafana and possible earth maps, automated data module tracker for Mark5 modules, dynamic ZABBIX maps, monitoring of additional systems using ZABBIX agents
 - software to extract monitoring data from log files as they are written
 - **dashboards/overviews of interest to different user groups**
 - scripts to upload data to centralized monitoring system in real time



European VLBI Network

Verification

Task 8.2

The Technische Universität München developed centralized monitoring system

- using industry standard basis software
 - time-series specialized database
 - large set of measurement points available “out of the box”
- continuation of previously EC-funded project in NEXPreS; currently available:
 - tools specific to VLBI stations
 - RFI Monitoring from spectra, mobile version of web pages, analysis of critical situations in the cryogenic Dewar, Grafana and possible earth maps, automated data module tracker for Mark5 modules, dynamic ZABBIX maps, monitoring of additional systems using ZABBIX agents
 - software to extract monitoring data from log files as they are written
 - dashboards/overviews of interest to different user groups
 - scripts to upload data to centralized monitoring system in real time



European VLBI Network

Verification

Task 8.2

“IVS Seamless Auxiliary Data Archive” at Wettzell Geodetic Observatory

- several years of experience built up in monitoring geodetic VLBI network
- detailed installation and usage instructions collected

D8.6: delivery of the production system at JIVE

- COVID-19 impact huge: ~1 yr extra delay in delivery
 - huge effort to move TUM to WfH operations
 - travel restrictions (could not travel to install system at JIVE)
- production system deployed at JIVE end of May 2021
- running w/o problems since then







European VLBI Network

Verification

Task 8.2

One of the TUM developed tools provides simple access to the monitoring data:

```
$ python3 ZabbixAPI.py -C ./config_evn.ini -L -H WETTZELL_000_NASAFIELDSystem -K ERC.DOTMON  
List of values of current item 28806 in current host: 10272
```

```
#Date                Unixtime      Value  
#-----  
2021-10-01 15:42:46  1633102966   7.9163  
2021-10-01 15:42:46  1633102966   7.9163  
2021-10-01 15:41:46  1633102906   7.9233  
2021-10-01 15:41:46  1633102906   7.9233  
2021-10-01 15:40:46  1633102846   7.9228  
2021-10-01 15:40:46  1633102846   7.9228  
2021-10-01 15:39:46  1633102786   7.9238  
2021-10-01 15:39:46  1633102786   7.9238
```



European VLBI Network

Verification

Task 8.2

One of the TUM developed tools provides simple access to the monitoring data:

```
$ python3 ZabbixAPI.py -C ./config_evn.ini -s system -K ERC.DOTMON
List of values of current item 28806

#Date                               Unixtime   Val
#-----
2021-10-01 15:42:46                 1633102966 7.91
2021-10-01 15:42:46                 1633102966 7.91
2021-10-01 15:41:46                 1633102906 7.9233
2021-10-01 15:41:46                 1633102906 7.9233
2021-10-01 15:40:46                 1633102846 7.9228
2021-10-01 15:40:46                 1633102846 7.9228
2021-10-01 15:39:46                 1633102786 7.9238
2021-10-01 15:39:46                 1633102786 7.9238
```

Anyone can run this and use the monitoring data!



European VLBI Network

Verification

Task 8.2

Impact on community difficult to assess:

- system only deployed few months ago, just before the holidays, after 2yr delay
- requires work by stations themselves
- stations are not partners in JUMPING JIVE WP8
 - effort not funded
 - JIVE is committed to provide support
- need to iterate with new user group
 - what is the useful information
 - what trends/alerts need to be set



European VLBI Network

Verification

Task 8.2

Impact on community difficult to assess:

- system only deployed few months ago, just before the holidays, after 2yr delay
- requires work by stations themselves
- stations are not partners in JUMPING JIVE WP8
 - effort not funded
 - JIVE is committed to provide support
- need to iterate with new user group
 - what is the useful information
 - what trends/alerts need to be set



European VLBI Network

Verification

Task 8.2

Impact on community difficult to assess:

- system only deployed few months ago, just before the holidays, after 2yr delay
- requires work by stations themselves
- stations are not partners in JUMPING JIVE WP8
 - effort not funded
 - **JIVE is committed to provide support**
- need to iterate with new user group
 - what is the useful information
 - what trends/alerts need to be set



European VLBI Network

Verification

Task 8.2

Impact on community difficult to assess:

- system only deployed few months ago, just before the holidays, after 2yr delay
- requires work by stations themselves
- stations are not partners in JUMPING JIVE WP8
 - effort not funded
 - JIVE is committed to provide support
- need to iterate with new user group
 - what is the useful information
 - what trends/alerts need to be set



ZABBIX Monitoring Inventory Reports

Dashboard **Problems** Overview Web Latest data Graphs Screens Maps Services

Problems Export to CSV

Filter

Show **Recent problems** Problems History

Host groups

Hosts

Application

Triggers

Problem

Minimum severity

Age less than days

Host inventory

Tags

Show tags Tag name

Tag display priority

Show suppressed problems Show unacknowledged only

Compact view Show timeline

Show details Highlight whole row

Show latest values

Time	<input type="checkbox"/>	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
2021-09-09 00:35:49	<input type="checkbox"/>	Warning		PROBLEM		WETTZELL_000_NASAFieldSystem	Data Archive: meteorological data are missing for 3 min.	20d 19h 12m	No		
2021-09-09 00:29:47	<input type="checkbox"/>	Information		PROBLEM		WETTZELL_000_NASAFieldSystem	Data Archive: wind data are missing for 3 min.	20d 19h 18m	No		

Displaying 2 of 2 found

0 selected



European VLBI Network

Verification

Task 8.2

Geodetic system (“IVS Seamless Auxiliary Data Archive”) used by geodetic community:

“Within the EMPIR project GeoMetre (18SIB01), the IVS Seamless Auxiliary Data Archive was used to get temperature data. These data were used to compensate for thermal expansions of the radio telescopes at Geodetic Observatory Wettzell. The simple data access to the archive makes our work more efficient.”



European VLBI Network

Verification

Task 8.2

Geodetic system (“IVS Seamless Auxiliary Data Archive”) used by geodetic community:

“Within the EMPIR project GeoMetre (18SIB01), the IVS Seamless Auxiliary Data Archive was used to get temperature data. These data were used to compensate for thermal expansions of the radio telescopes at Geodetic Observatory Wettzell. The simple data access to the archive makes our work more efficient.”



European VLBI Network

Verification

Task 8.2

Geodetic system (“IVS Seamless Auxiliary Data Archive”) used by geodetic community:

“The IVS Seamless Auxiliary Data Archive is an excellent tool to retrieve continuous information from meteorological sensors and clocks. The temperature values, for example, are important to determine accurate parameters for the thermal deformation of the radio telescopes. Continuous pressure values on the other hand allow for a rigorous comparison against pressure values from other sources, such as numerical weather models, as needed for troposphere delay modeling”



European VLBI Network

Verification

Task 8.2

Geodetic system (“IVS Seamless Auxiliary Data Archive”) used by geodetic community:

“The IVS Seamless Auxiliary Data Archive is an excellent tool to retrieve continuous information from meteorological sensors and clocks. The temperature values, for example, are important to determine accurate parameters for the thermal deformation of the radio telescopes. Continuous pressure values on the other hand allow for a rigorous comparison against pressure values from other sources, such as numerical weather models, as needed for troposphere delay modeling”



European VLBI Network

Verification

Task 8.2

Objective:

“The final product will be a central, web-based monitoring system, usable for both astronomical and geodetic VLBI. This system will be accessible to all involved, correlator and stations, and will serve to continuously monitor and assess the status of the VLBI network, enabling automated warnings in case of failures and providing the information needed to continuously improve the performance of the network.”



Summary of measured impact on VLBI arrays such as the EVN

Expert as well as novice users appreciate the modernized pySCHED a lot

- lower the barrier to entry into scheduling a VLBI observation
- virtually no installation problems anymore
- automatic updates save time and loss of science due to scheduling problems
- plotting has improved a lot - easier to verify/inspect the configuration

A centralized monitoring system is shown to provide

- insight in, and, *real-time* overview of system status improves reliability
- history of monitoring data improves efficiency
 - can look back in time to find when problem starts/ends
- instant error or warning triggers shortens diagnosis timescale by order of magnitude
- access for external users allows improvement of calibration, i.e. science potential