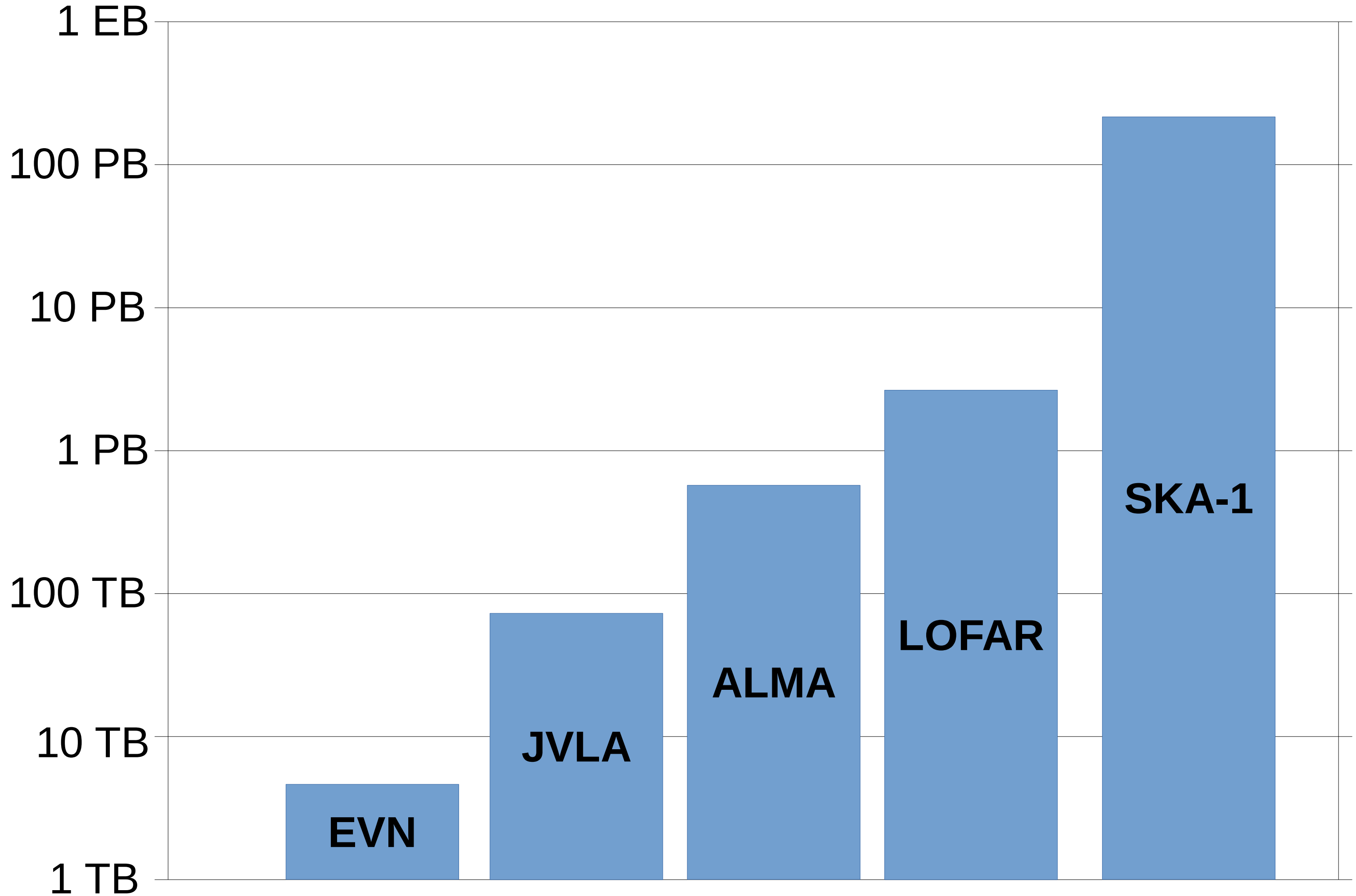
Integration of CASA with Jupyter for efficient remote processing

Aard Keimpema *(keimpema@jive.eu)*

H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).

Yearly archivable data



Near data processing

- SKA phase-1 will produce ~1 PB / day
- Data reduction will be done where the data is stored
- Possible solution: Remote interactive pipelines based on Jupyter notebooks
- Jupyter notebooks are displayed inside a web-browser
- Successor to IPython, CASA is based on IPython
- Not limited to python, bindings to 40+ languages exist

jupyter vla-cont-tutorial Last Checkpoint: 06/01/2017 (autosaved)

VLA Continuum Tutorial 3C391-CASA4.7

Source : https://casaguides.nrao.edu/index.php/VLA_Continuum_Tutorial_3C391-CASA4.7

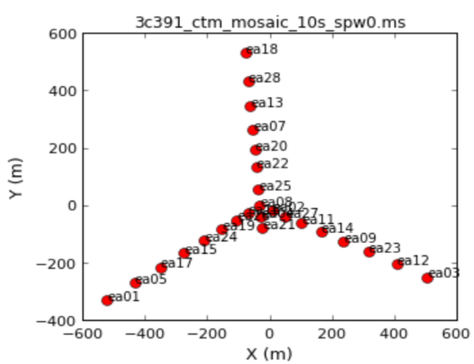
Download the data from the above location and extract the dataset to the same directory as this notebook is located

In [2]:
```
listobs(vis='3c391_ctm_mosaic_10s_spw0.ms')
```

```
================================================================================
           MeasurementSet Name:  /home/jupyter/data/casa/3c391_ctm_mosaic_10s_spw0.ms      MS Version 2
================================================================================
   Observer: Dr. James Miller-Jones     Project: T.B.D.
Observation: EVLA
Data records: 845379       Total elapsed time = 28681.5 seconds
   Observed from   24-Apr-2010/08:02:10.0   to   24-Apr-2010/16:00:11.5 (UTC)

   ObservationID = 0         ArrayID = 0
  Date        Timerange (UTC)          Scan  FldId FieldName             nRows     SpwIds   Average Interval(s)
ScanIntent
  24-Apr-2010/08:02:10.0 - 08:02:30.0     1      0 J1331+3030              650  [0]  [10]
              08:02:20.0 - 08:09:30.0     2      0 J1331+3030            13975  [0]  [10]
              08:09:20.0 - 08:16:28.0     3      0 J1331+3030            13975  [0]  [10]
              08:19:38.0 - 08:24:26.5     4      1 J1822-0938             7035  [0]  [10]
              08:24:48.0 - 08:29:48.0     5      2 3C391 C1               7590  [0]  [10]
              08:29:38.0 - 08:34:48.0     6      3 3C391 C2               7821  [0]  [10]
              08:34:38.0 - 08:39:48.0     7      4 3C391 C3               7821  [0]  [10]
              08:39:38.0 - 08:44:48.0     8      5 3C391 C4               7821  [0]  [10]
              08:44:38.0 - 08:49:48.0     9      6 3C391 C5               7843  [0]  [10]
```

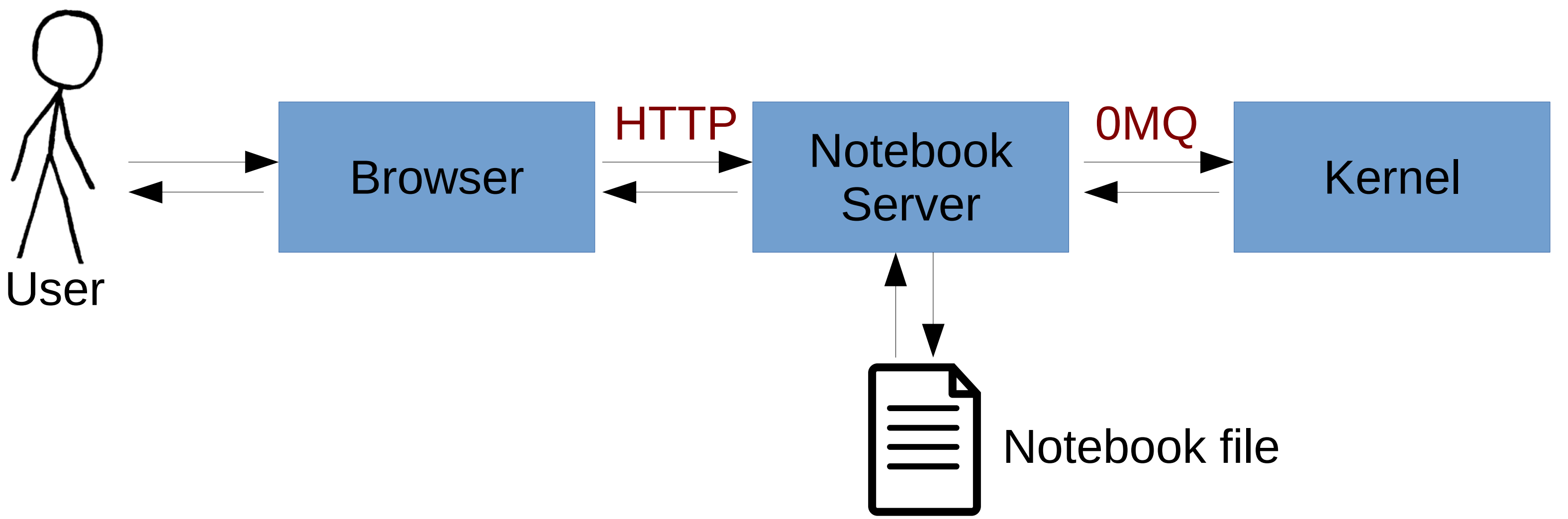
In [3]:
```
plotants(vis='3c391_ctm_mosaic_10s_spw0.ms',figfile='plotants_3c391_antenna_layout.png')
clearstat() # This removes the table lock generated by plotants in script mode
```



Show log

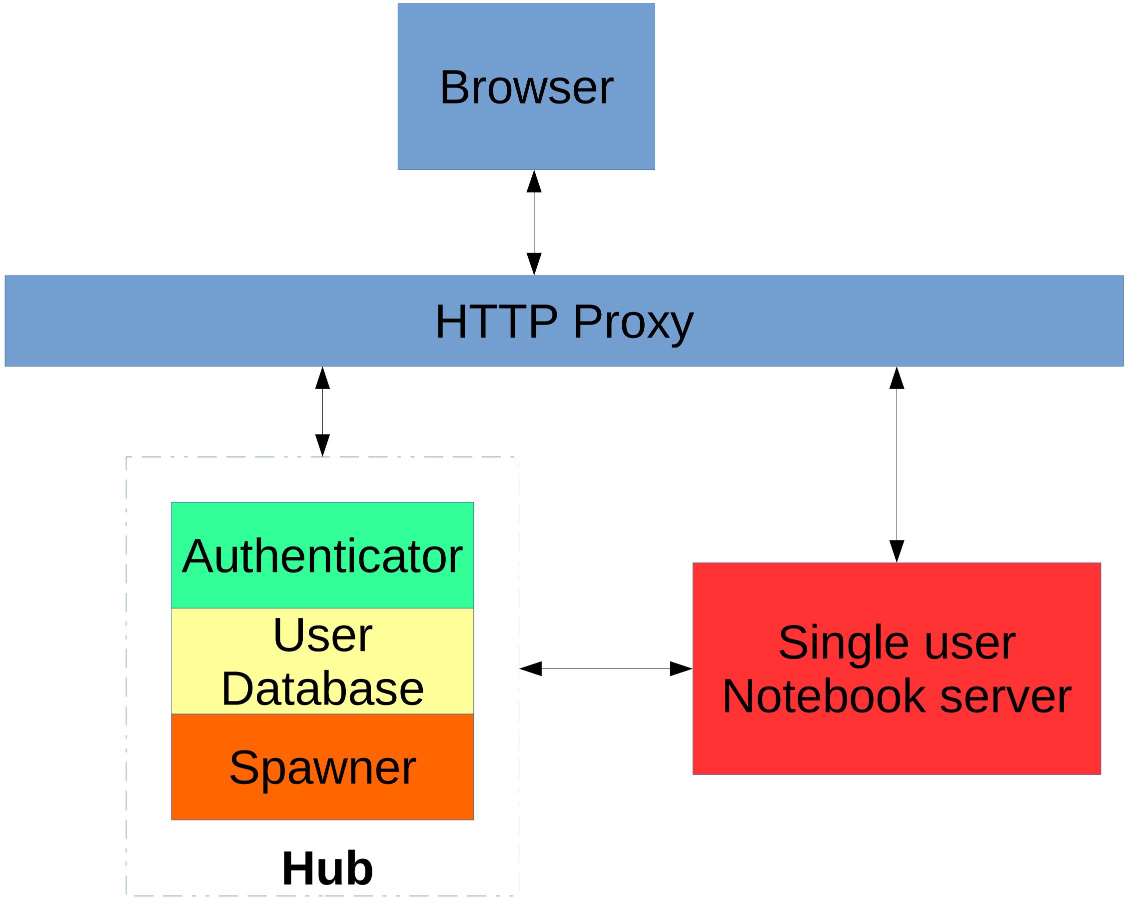
In [5]:
```
flagdata(vis='3c391_ctm_mosaic_10s_spw0.ms', flagbackup=T,
```

Jupyter Architecture



- Notebook server knows nothing about target language
- All language specifics are in the kernel
  - *Wrapper-kernel*: written in python, easiest to implement
  - *Native-kernel*: written in target language, much more work

Jupyter hub

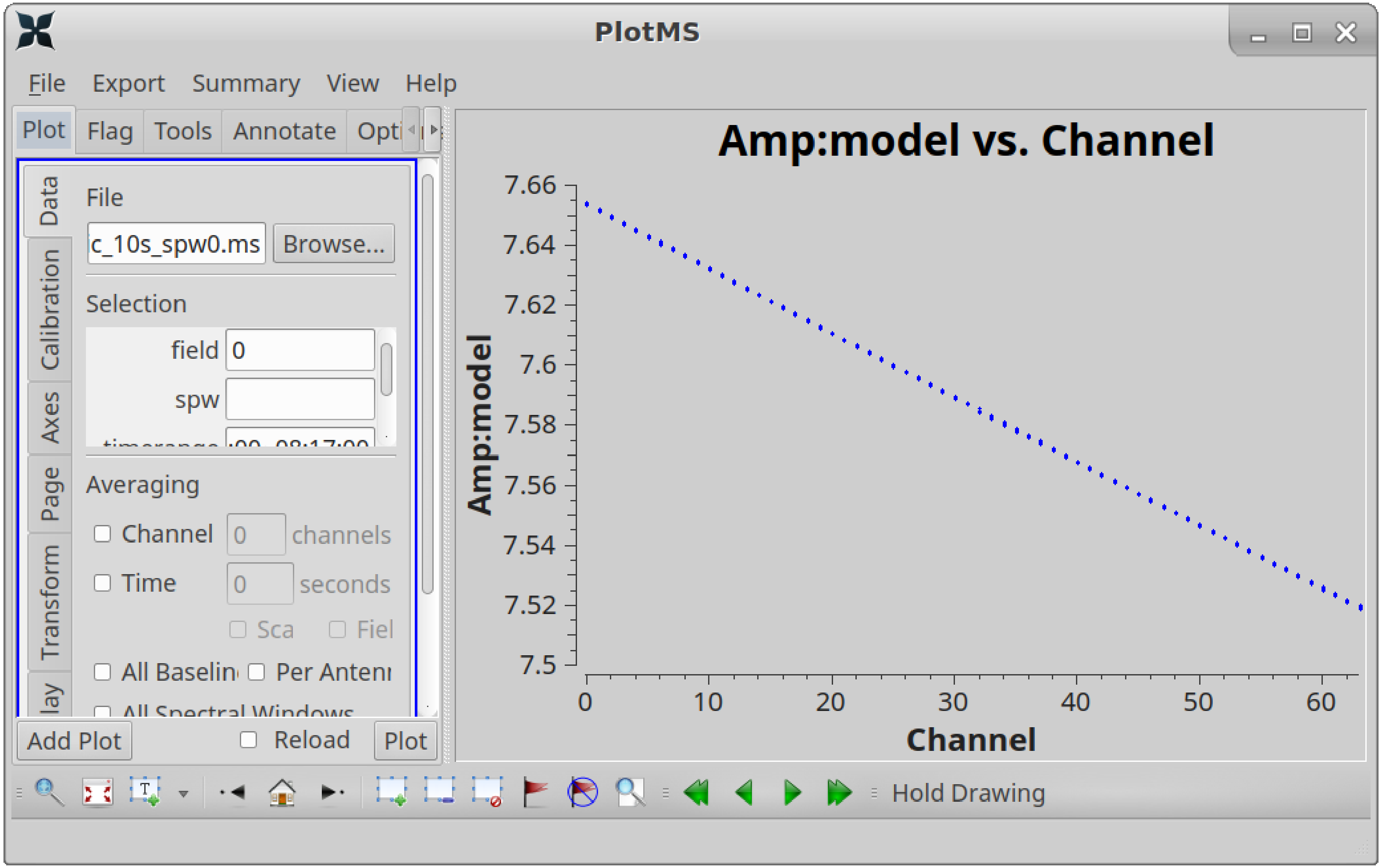


Multi-user front-end for Jupyter

CASA Jupyter kernel

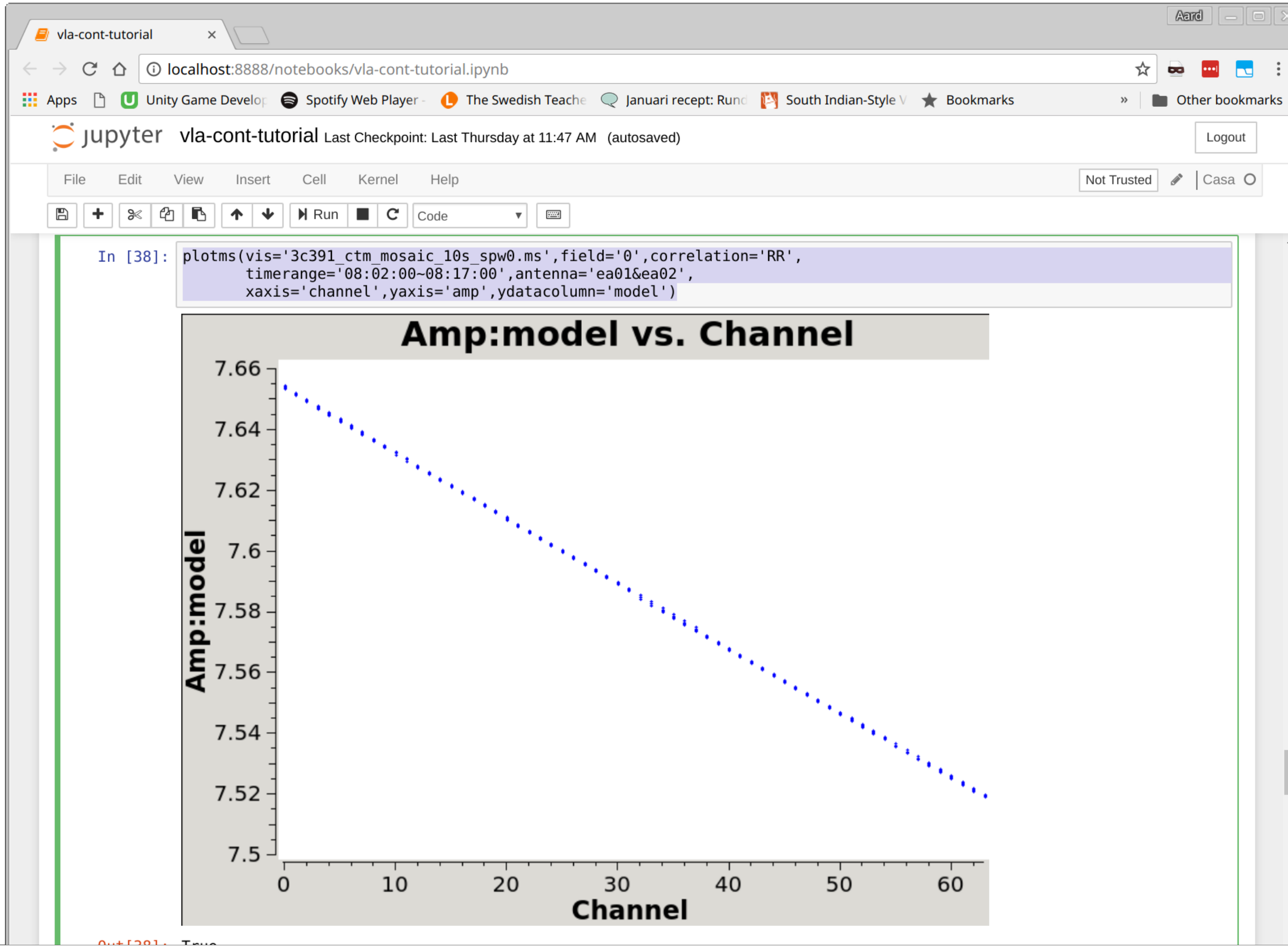
- Based on the generic python wrapper kernel
- Initialization:
  - Load needed python packages: casacore, casa tasks, matplotlib, ….
  - Setup environment: Config, logging, dbus, etc..
- CASA has python bindings for all tasks
- Many tasks open a C++ coded GUI, these are wrapped so that output goes to notebook.
- *https://github.com/aardk/jupyter-casa*

Example: plotms



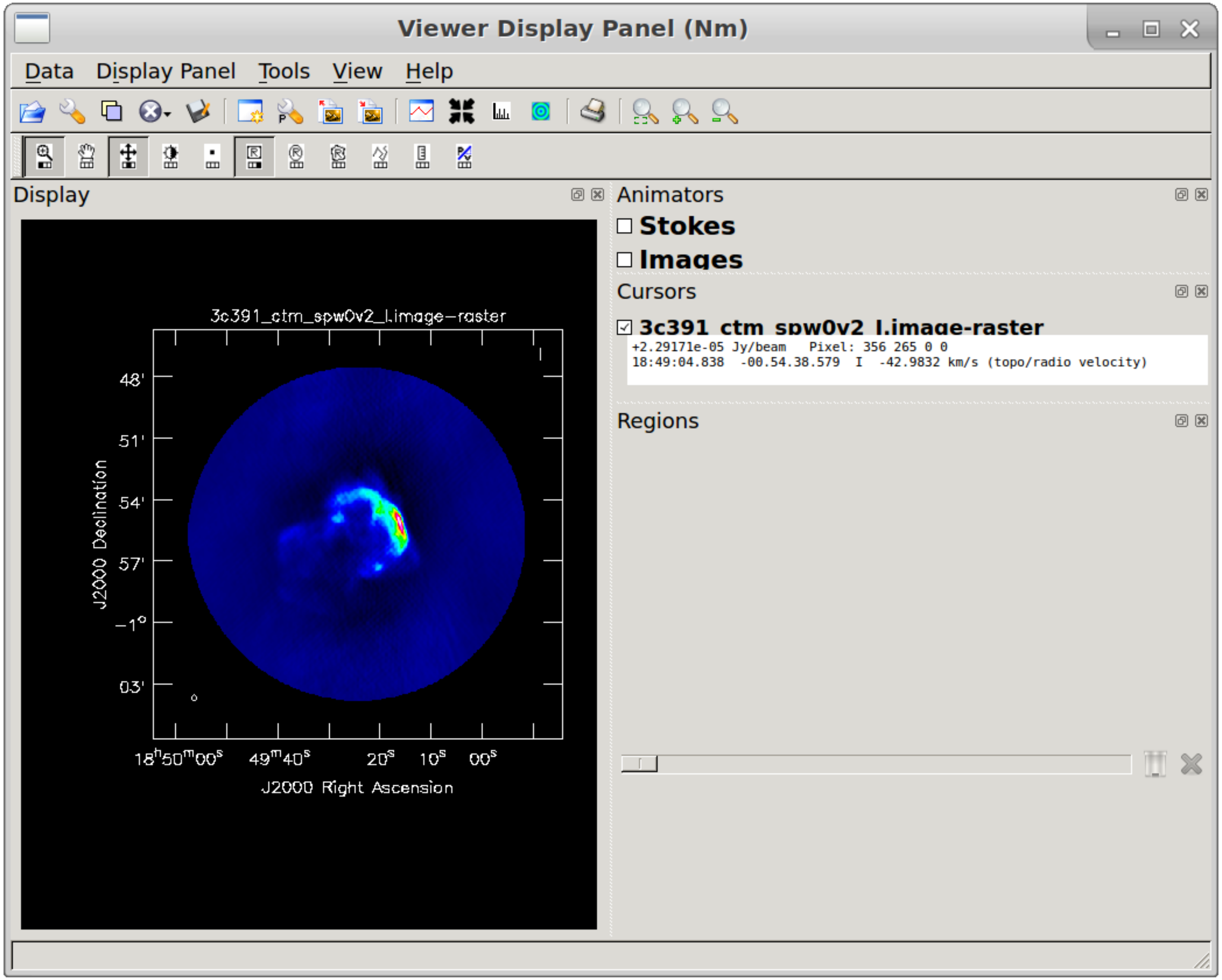

```
plotms(vis='3c391_ctm_mosaic_10s_spw0.ms', field='0', correlation='RR',
    timerange='08:02:00~08:17:00', antenna='ea01&ea02',
    xaxis='channel', yaxis='amp', ydatacolumn='model')
```

Example: plotms



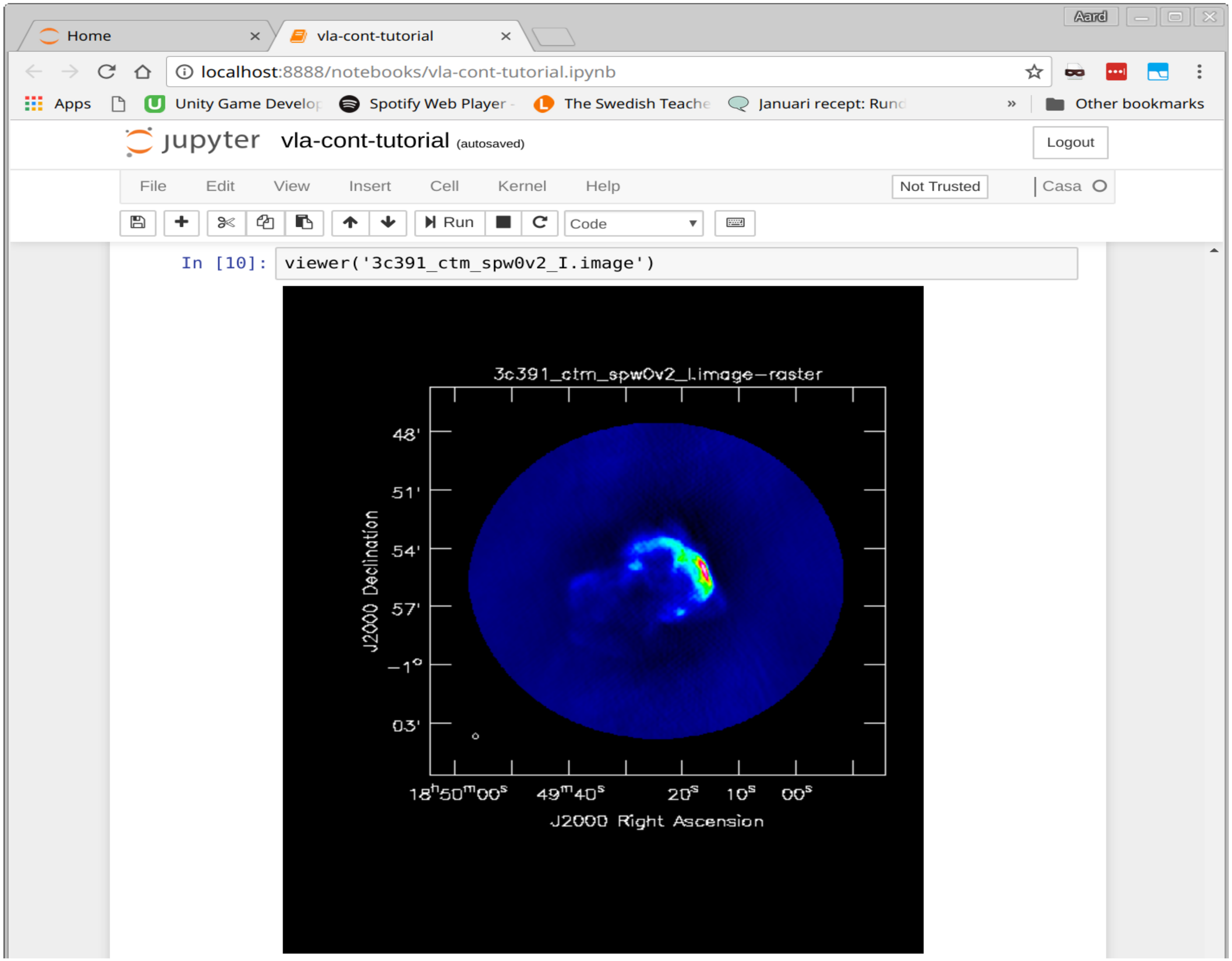
plotms(vis='3c391_ctm_mosaic_10s_spw0.ms', field='0', correlation='RR',
… , xaxis='channel', yaxis='amp', ydatacolumn='model',
plotfile='plotms_temp.png', showgui=False)

Example: casaviewer



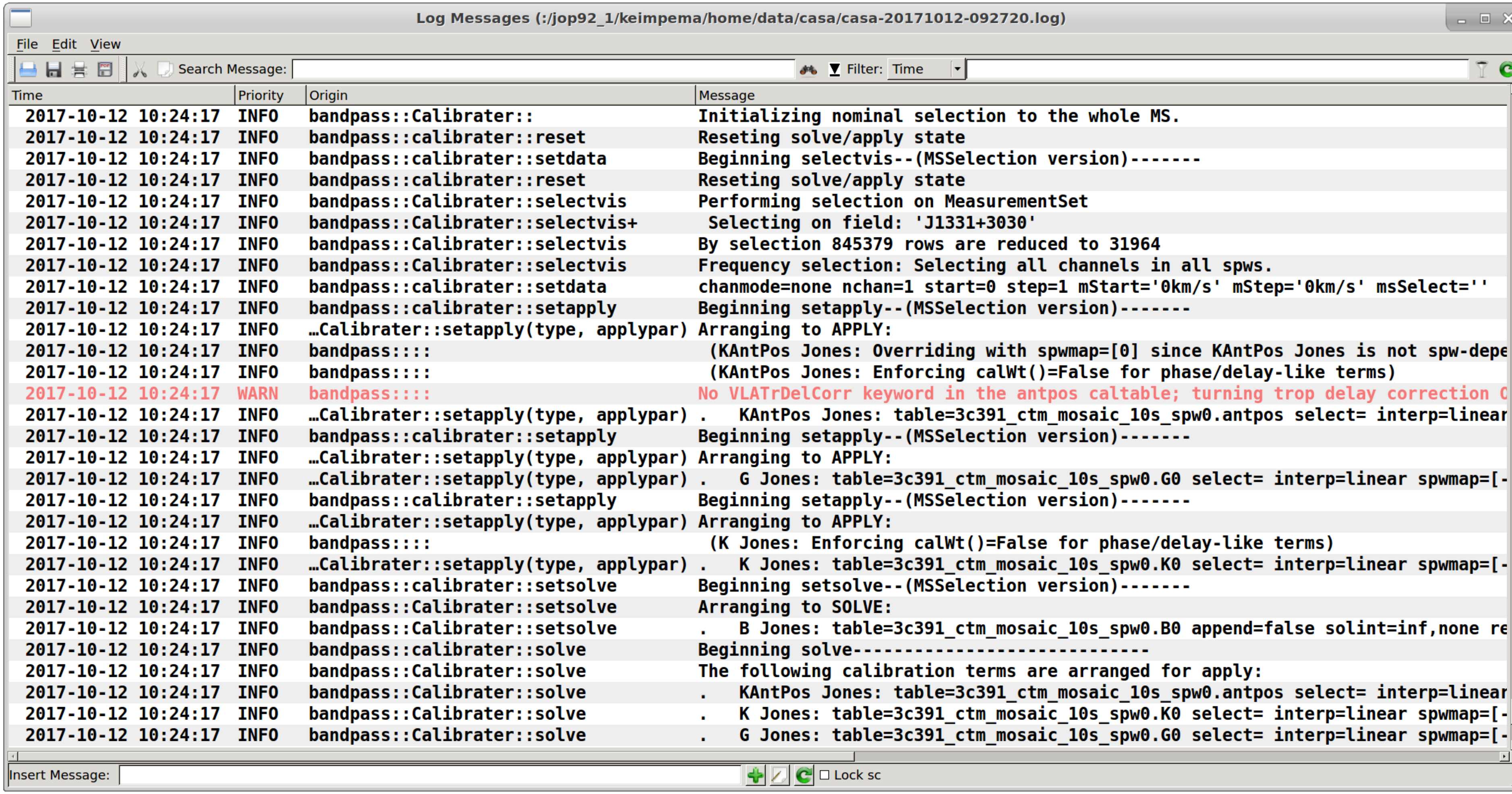
viewer('3c391_ctm_spw0v2_I.image')

Example: casaviewer



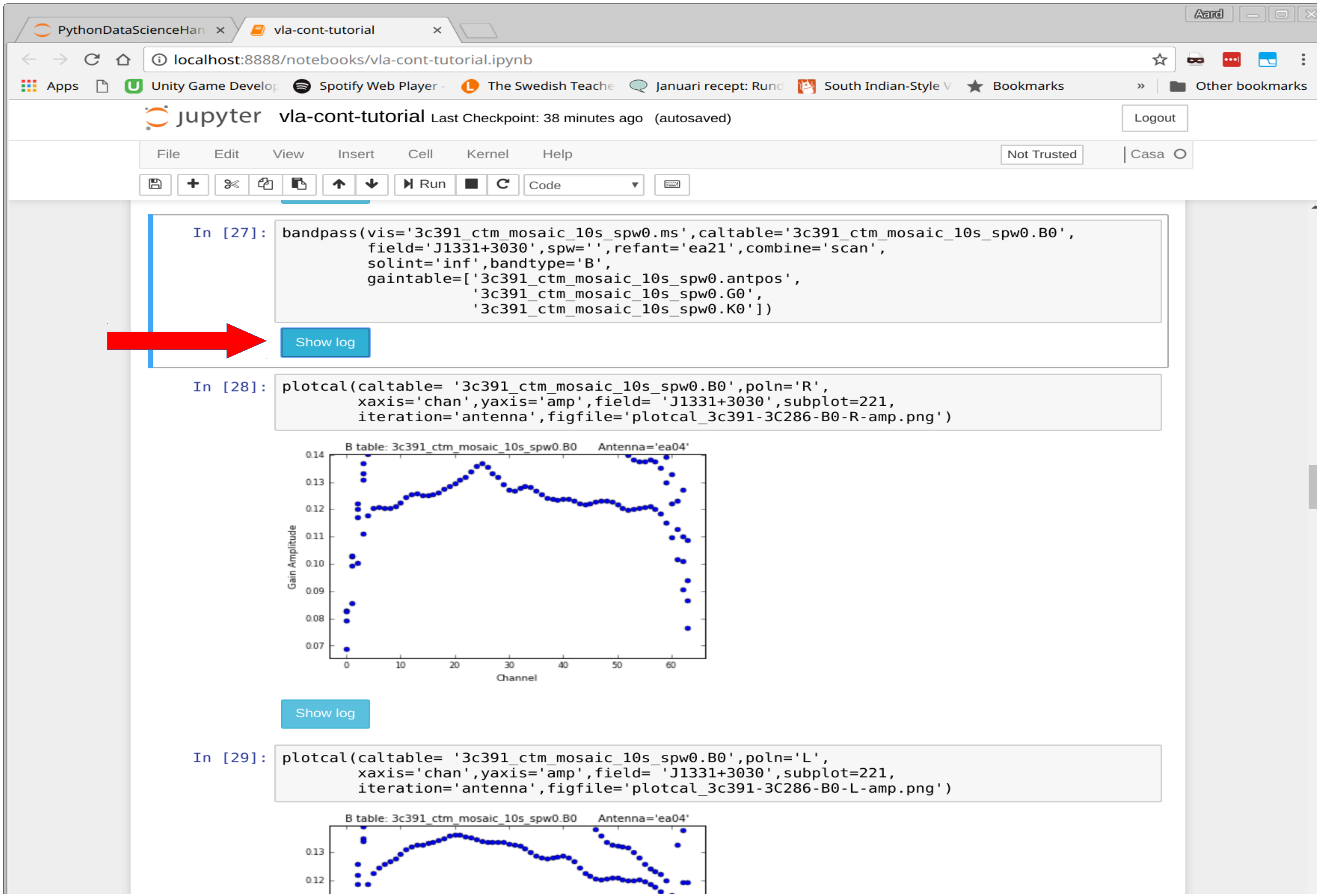
viewer('3c391_ctm_spw0v2_I.image', **gui = False**, **outformat = 'png'**, **outfile = viewer_temp.png**)

Logging



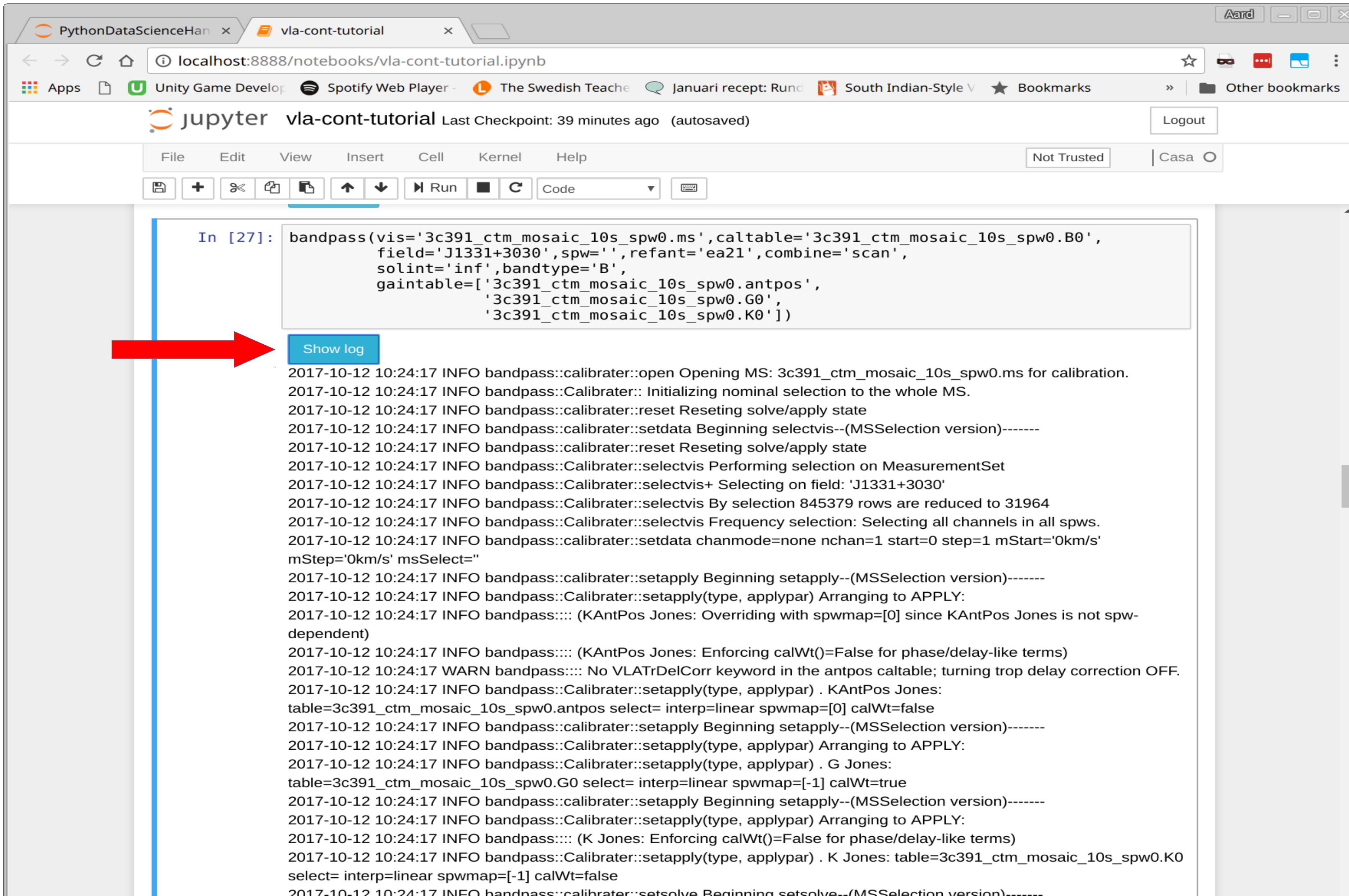
CASA displays logging information inside *casalogger* task.

Logging



PythonDataScienceHan
vla-cont-tutorial
localhost:8888/notebooks/vla-cont-tutorial.ipynb
Apps
Unity Game Develo
Spotify Web Player
The Swedish Teache
Januari recept: Run
South Indian-Style
Bookmarks
Other bookmarks
jupyter
vla-cont-tutorial Last Checkpoint: 38 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Not Trusted
Casa
Run
Code
In [27]:
bandpass(vis='3c391_ctm_mosaic_10s_spw0.ms',caltable='3c391_ctm_mosaic_10s_spw0.B0',
field='J1331+3030',spw='',refant='ea21',combine='scan',
solint='inf',bandtype='B',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos',
'3c391_ctm_mosaic_10s_spw0.G0',
'3c391_ctm_mosaic_10s_spw0.K0'])
Show log
In [28]:
plotcal(caltable= '3c391_ctm_mosaic_10s_spw0.B0',poln='R',
xaxis='chan',yaxis='amp',field= 'J1331+3030',subplot=221,
iteration='antenna',figfile='plotcal_3c391-3C286-B0-R-amp.png')
B table: 3c391_ctm_mosaic_10s_spw0.B0
Antenna='ea04'
Gain Amplitude
Channel
Show log
In [29]:
plotcal(caltable= '3c391_ctm_mosaic_10s_spw0.B0',poln='L',
xaxis='chan',yaxis='amp',field= 'J1331+3030',subplot=221,
iteration='antenna',figfile='plotcal_3c391-3C286-B0-L-amp.png')
B table: 3c391_ctm_mosaic_10s_spw0.B0
Antenna='ea04'

Logging


PythonDataScienceHan
vla-cont-tutorial
localhost:8888/notebooks/vla-cont-tutorial.ipynb
Apps
Unity Game Develop
Spotify Web Player
The Swedish Teache
Januari recept: Runc
South Indian-Style V
Bookmarks
Other bookmarks
jupyter vla-cont-tutorial Last Checkpoint: 39 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Not Trusted
Casa
Run
Code
In [27]:
bandpass(vis='3c391_ctm_mosaic_10s_spw0.ms',caltable='3c391_ctm_mosaic_10s_spw0.B0',
field='J1331+3030',spw='',refant='ea21',combine='scan',
solint='inf',bandtype='B',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos',
'3c391_ctm_mosaic_10s_spw0.G0',
'3c391_ctm_mosaic_10s_spw0.K0'])
Show log
2017-10-12 10:24:17 INFO bandpass::calibrater::open Opening MS: 3c391_ctm_mosaic_10s_spw0.ms for calibration.
2017-10-12 10:24:17 INFO bandpass::Calibrater:: Initializing nominal selection to the whole MS.
2017-10-12 10:24:17 INFO bandpass::calibrater::reset Reseting solve/apply state
2017-10-12 10:24:17 INFO bandpass::calibrater::setdata Beginning selectvis--(MSSelection version)-------
2017-10-12 10:24:17 INFO bandpass::calibrater::reset Reseting solve/apply state
2017-10-12 10:24:17 INFO bandpass::Calibrater::selectvis Performing selection on MeasurementSet
2017-10-12 10:24:17 INFO bandpass::Calibrater::selectvis+ Selecting on field: 'J1331+3030'
2017-10-12 10:24:17 INFO bandpass::Calibrater::selectvis By selection 845379 rows are reduced to 31964
2017-10-12 10:24:17 INFO bandpass::Calibrater::selectvis Frequency selection: Selecting all channels in all spws.
2017-10-12 10:24:17 INFO bandpass::calibrater::setdata chanmode=none nchan=1 start=0 step=1 mStart='0km/s' mStep='0km/s' msSelect=''
2017-10-12 10:24:17 INFO bandpass::calibrater::setapply Beginning setapply--(MSSelection version)-------
2017-10-12 10:24:17 INFO bandpass::Calibrater::setapply(type, applypar) Arranging to APPLY:
2017-10-12 10:24:17 INFO bandpass:::: (KAntPos Jones: Overriding with spwmap=[0] since KAntPos Jones is not spw-dependent)
2017-10-12 10:24:17 INFO bandpass:::: (KAntPos Jones: Enforcing calWt()=False for phase/delay-like terms)
2017-10-12 10:24:17 WARN bandpass:::: No VLATrDelCorr keyword in the antpos caltable; turning trop delay correction OFF.
2017-10-12 10:24:17 INFO bandpass::Calibrater::setapply(type, applypar) . KAntPos Jones: table=3c391_ctm_mosaic_10s_spw0.antpos select= interp=linear spwmap=[0] calWt=false
2017-10-12 10:24:17 INFO bandpass::calibrater::setapply Beginning setapply--(MSSelection version)-------
2017-10-12 10:24:17 INFO bandpass::Calibrater::setapply(type, applypar) Arranging to APPLY:
2017-10-12 10:24:17 INFO bandpass::Calibrater::setapply(type, applypar) . G Jones: table=3c391_ctm_mosaic_10s_spw0.G0 select= interp=linear spwmap=[-1] calWt=true
2017-10-12 10:24:17 INFO bandpass::calibrater::setapply Beginning setapply--(MSSelection version)-------
2017-10-12 10:24:17 INFO bandpass::Calibrater::setapply(type, applypar) Arranging to APPLY:
2017-10-12 10:24:17 INFO bandpass:::: (K Jones: Enforcing calWt()=False for phase/delay-like terms)
2017-10-12 10:24:17 INFO bandpass::Calibrater::setapply(type, applypar) . K Jones: table=3c391_ctm_mosaic_10s_spw0.K0 select= interp=linear spwmap=[-1] calWt=false
2017-10-12 10:24:17 INFO bandpass::calibrater::setsolve Beginning setsolve--(MSSelection version)-------

CASA for Jupyter

- NRAO CASA distribution is entirely self-contained, it is essentially a linux distribution
- Too old for Jupyter, many conflicting packages, e.g. Matplotlib, IPython, ….
- We created a custom build of CASA using based on latest Python
- Only functional with Jupyter, no stand-alone CASA interpreter.
- Distributed as **Docker** and **Singularity** containers.

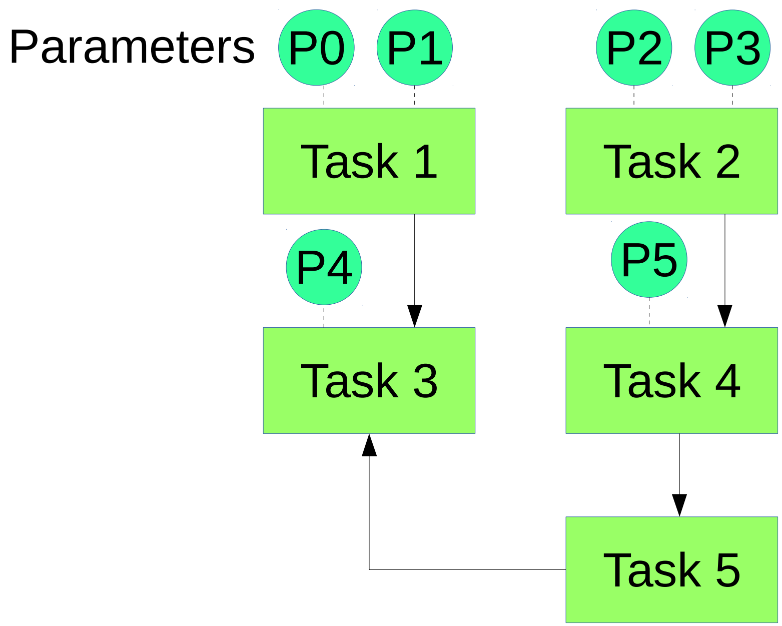
Docker containers

- Lightweight alternative to virtual machines
- Docker containers are isolated from host, resources have to be shared explicitly.
- Download Jupyter-CASA from dockerhub:
  - *docker pull penngwyn/jupytercasa*
- Run Jupyter-CASA:
  - *docker run --rm -p 8888:8888 -i -t -v /tmp/.X11-unix:/tmp/.X11-unix -e DISPLAY=$DISPLAY penngwyn/jupytercasa /bin/sh -c "jupyter notebook"*
- Docker not acceptable at many computing centres: Root privilege escalation

Singularity containers

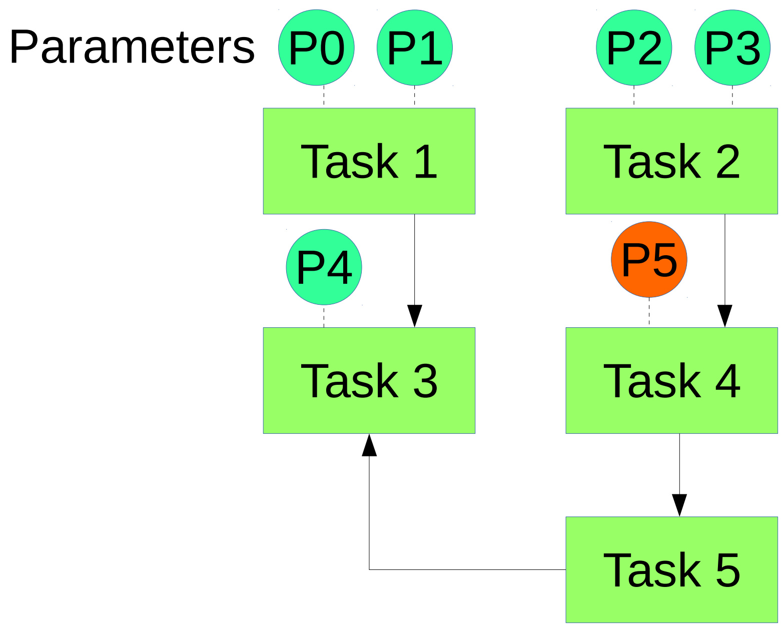
- Can import directly from Docker
- Less isolated: /dev, /tmp, /proc, and $HOME are shared with host
- Runs as in the user context of the user that executes: “singularity run”
- Download Jupyter-CASA from Singularity-hub
  - *singularity pull shub://aardk/jupyter-casa*
- Run Jupyter-CASA:
  - *singularity run aardk-jupyter-casa-master.img*
- Doesn’t have root privilege escalation issues

Minimal re-computation pipelines



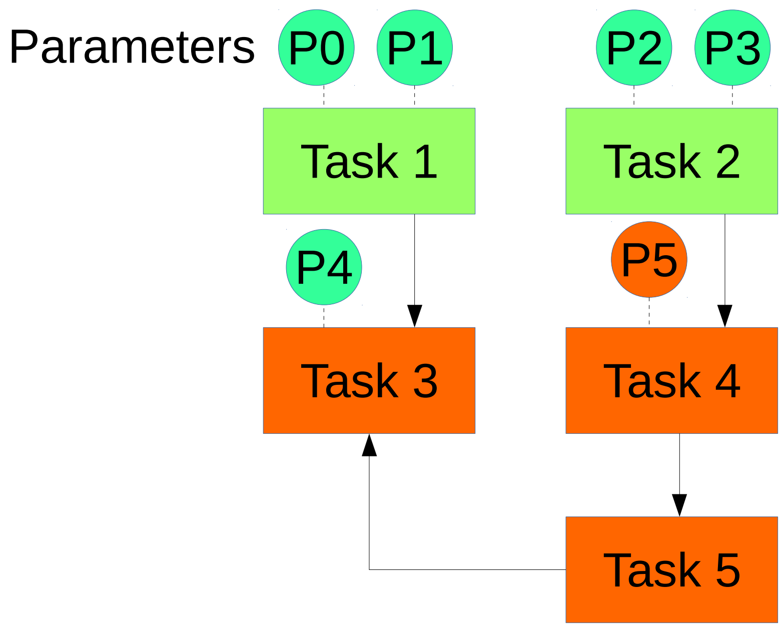
- Integrate minimal re-computation into Jupyter
- Original framework part of Hilado / Radionet
  - JIVE: Des Small and Mark Kettenis
  - U. Cambridge: Bojang Nikolic

Minimal re-computation pipelines



- Integrate minimal re-computation into Jupyter
- Original framework part of Hilado / Radionet
  - JIVE: Des Small and Mark Kettenis
  - U. Cambridge: Bojang Nikolic

Minimal re-computation pipelines



- Integrate minimal re-computation into Jupyter
- Original framework part of Hilado / Radionet
  - JIVE: Des Small and Mark Kettenis
  - U. Cambridge: Bojang Nikolic

Conclusions

- We have implemented Jupyter kernel for CASA suitable for pipelines
- We provide both Docker and Singularity images for easy deployment
- Future work includes:
  - Implement minimal re-computation
  - Jupyter hub integration
- *https://github.com/aardk/jupyter-casa*

Text for acknowledgement Slide

Acknowledgement

- H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).