System Agnostic Scripting for Radio Interferometry

Simulate
Calibrate
Image

stimela

Sphesihle Makhathini
SKA SA, Rhodes University
smakhathini@ska.ac.za
The data is bigly

- More data than we can process, with a lot more on the way
- Instruments and sky more complex
- Data reduction, analysis and pipelining tools increasingly more complex
  a. Poorly documented
  b. Unstable
  c. Difficult to build/install
- Astronomers write bad code (#NotAllAstromers)
  a. It needs to work for this case, so I can get my PhD/paper result
- Increasingly difficult to reproduce science data products

To meet expectations of modern instruments

- Flexible, easy to use, robust, scalable, easily portable workflow management tools
Packaging

Packaging software is vital to producing reproducible science data products

- Manage dependencies
- Proper versioning of software
- Forces basic level of standardization of how you write software
- Lowers burden on user
- Many tools to help you do it (even automate it)

➔ Launchpad: https://launchpad.net
➔ Python Packaging Index (pip): https://pypi.python.org/pypi
➔ Kernsuite (For radio astronomy packages): http://kernsuite.info
Contact Kernsuite

$ apt-get install 21cmfast simfast21

Ensure that the code will at least install correctly; with proper management of deps.

Only supports Ubuntu LTS distros

https://github.com/kernsuite/packaging
Container Technology

- OS level virtualization
- Build self contained computing environments for applications
- Minimal interaction with host system (only share Linux Kernel)
- Lightweight, and very little overhead
- Presents an elegant and simple solution to most problems with RI software applications.
- Build. Ship. Run

coreos.com/rkt

singularity.lbl.gov

docker.com
Docker

```
FROM ubuntu:16.04

# Adding the kernsuite PPA
RUN apt-get update
RUN apt-get install software-properties-common -y
RUN apt-add-repository -s ppa:kernsuite/kern-1
RUN apt-add-repository multiverse

# Installing wsclean
RUN apt-get update && \
    apt-get install -y \
    wsclean
```

Pull from docker hub (https://hub.docker.com)

$ docker build -t wsclean <location of Dockerfile>
$ docker run wsclean /usr/bin/wsclean <args>

Commit changes and push new image

$ docker commit <container name/ID> <image name>
$ docker push <new image name>
Stimela

- Combines Docker & Kernsuite to create a platform agnostic scripting framework.
- Provides easy access to legacy and novel applications
- Simple Python scripting environment
- Modular, Portable, Robust

[https://github.com/SpheMakh/Stimela](https://github.com/SpheMakh/Stimela)
[https://github.com/SpheMakh/Stimela/wiki](https://github.com/SpheMakh/Stimela/wiki)

$ pip install stimela
Architecture

Python UI
Scripting environment

Python

Kernsuite

CASA
WSClean
AOFlager
LOFARSoft
SIMMS
MeqTrees
LWImage
Tigger
+++
Workflow of a stimela module (a.k.a cab)

Task
(Imaging/Calibration/flagging)

Input
Parameters
Output
Log file

```json
{
  "task": "casa_uvsub",
  "base": "stimela/casa",
  "tag": "0.2.5",
  "description": "Subtract/add model from/to the corrected visibility data."
}

[parameters:]

- "info": "Name of input visibility file (MS)",
- "name": "msname",
- "io": "msfile",
- "default": null,
- "dtype": "file",
- "mapping": "vis"

[info]: "reverse the operation (add rather than subtract)"
- "dtype": "bool",
- "default": false,
- "name": "reverse"
```
Example

```python
import stimela

# recipe I/O flow
INPUT = "input"
OUTPUT = "output"
MSDIR = "msdir"

MS = "measurement_set_name"
LSM = "sky_model_name"

# Start recipe instance
recipe = stimela.Recipe("Example Stimela Calibration Recipe", ms_dir=MSDIR)
```
Example

```python
# Add task to the recipe
recipe.add("cab/calibrator", "Gjones_cal_1", # Label of container that will be created from the Docker image
{ # Parameters to be parsed to task
  "msname" : MS,
  "skymodel" : LSM,
  "Gjones" : True,
  "Gjones_solution_intervals" : [1,1],
},
    input=INPUT, output=OUTPUT, # Task I/O
    label="Direction independent calibration", # Its good to log what you doing
)
```python
recipe.add("cab/wsclean", "make_image", {
    "msname": MS,
    "prefix": "example_stimela_calibration",
    "npx": 4096,
    "cellsize": 2, # in arcsec
    "weight": "briggs 0",
},

input=INPUT, output=OUTPUT,
label="Image visibility data",
}
```

```bash
$ stimela run <recipe name>
```

```python
recipe.run()
# recipe.run([1,2])
# recipe.run([1])
# recipe.run([2])
# recipe.run([2,1])
```
VerMeerKAT

End-to-end MeerKAT data reduction pipeline
- Full treatment of DDEs
- Portable, configurable
- Can be easily deployed on any system (with Docker)

Install and run
$ pip install git+https://github.com/ska-sa/vermeerkat
$ vermeerkat -f <observation id>.h5 -c <config file>

https://github.com/ska-sa/vermeerkat (private for now)
VerMeerKAT continuum pipeline

https://github.com/ska-sa/vermeerkat (private for now)
DEEP2

Amp:corrected vs. Frequency Corr: XX

Phase:corrected vs. Frequency Corr: XX
DEEP2
Conclusion

- Document your code
- Package it
- Try out container technology
- We have the tools, expertise and resources to make robust, modular and scalable data reduction/simulation pipelines.
I mailed you the code

Where are the cmake/setup.py files!?