ilifu Online Training – Session 4

Walter Silima

ilifu Astronomy Support Specialist
IDIA, Department of Astronomy, University of Cape Town

IDIA Inter-University Institute for Data Intensive Astronomy

UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

ilifu

Data Transfers

- http://docs.ilifu.ac.za/#/data/data_transfer
- **Do not use the login node (**slurm.ilifu.ac.za**)!**
- Use other services such as transfer.ilifu.ac.za node or Globus
- **transfer.ilifu.ac.za node:**
  - For internal and external copying of data (cp, scp, rsync, etc)
  - Useful for transferring data up to 200 GB
  - For example:
    - `$ cp -a /users/$USER/scripts/ /idia/users/$USER/`
    - `$ scp /path/to/file/<filename> <username>@transfer.ilifu.ac.za:/idia/users/$USER/scripts/`
    - `$ rsync /path/to/file/<filename> <username>@transfer.ilifu.ac.za:/idia/users/$USER/scripts/`
    - read the manual pages using **man scp** or **man rsync**

Data Transfers

- **Globus**
  - files over 200 GB or for frequent transfers
  - Need two well configured end-points to make a transfer
  - Much faster than scp and rsync
    - A few to 10 Gbps locally (e.g. SARAO archive, CHPC, local server)
    - 1.2 Gbps to Perth, 5 Gbps to the Netherlands (compared to 10s Mbps)
  - Uses dedicated data transfer node (DTN)
  - Checks file integrity and includes similar functionality to rsync
  - **Please note that large files should not be copied into your /users/ directory**

Data Transfers: Globus

- http://docs.ilifu.ac.za/#/data/data_transfer?id=transfer-using-globus-online (two authentication steps)
- User-friendly globus connect built on top of GridFTP
  - GUI/web app or CLI
- Can connect any arbitrary end points
  - Server (DTN), desktop, etc for Mac OS X, Windows & Linux
  - Offers user-friendly but computationally efficient transfer service
- Globus recommended for larger or more frequent transfers
  - Run background job from user-friend web app with file browser
  - Avoids fiddling with `scp/rsync`, keeping active connection, and long wait times

Data Transfers: Globus

- Symbolic links (symlinks) aren't included
  - http://docs.ilifu.ac.za/#/data/data_transfer?id=configuring-a-transfer
  - e.g. CASA Multi-MS (MMS)
  - Can be repaired with `rsync` and "preserve source file modification times"
- Demo!

Data Management

- ilifu is a shared resource-limited cluster
- is shared amongst all members of our user community
- Supports a diverse range of projects
  - Astronomy and Bioinformatics
  - Varying resource requirements
- Efficient use of storage essential
  - Achieved via a data management plan
  - Laid out in data management guide
    - Strategies, best practices and workflows

A good data management plan

Prototype

- Prototype your workflow over small volumes
  - via a version-controlled repository

Develop

- Develop workflow into fully-automated production workflow

Write

- Automatically write selected products to longer-term storage
  - Including logs, software versions and input parameters

Remove

- Automatically remove temporary/intermediate data products (i.e. the remainder)

Typical workflow

As detailed in Directory structure:

- The purpose of ***scratch*** mounts is data processing
  - expected to contain temporary data that can be quickly removed
- The Project directories **/{idia,cbio,ilifu}/projects/** are project-specific directories
  - expected to contain final data products for longer-term storage
- The **/users** mounts is meant to store scripts and config files
- **A good Workflow**
  - Utilises this structure
  - **/users** used to run processing workflow or pipeline
  - Reading (e.g. raw) data from read-only directory
    - `/n/raw, /n/projects or /n/data`
  - Writing temporary/intermediate data products to scratch mount
    - `/scratch3`

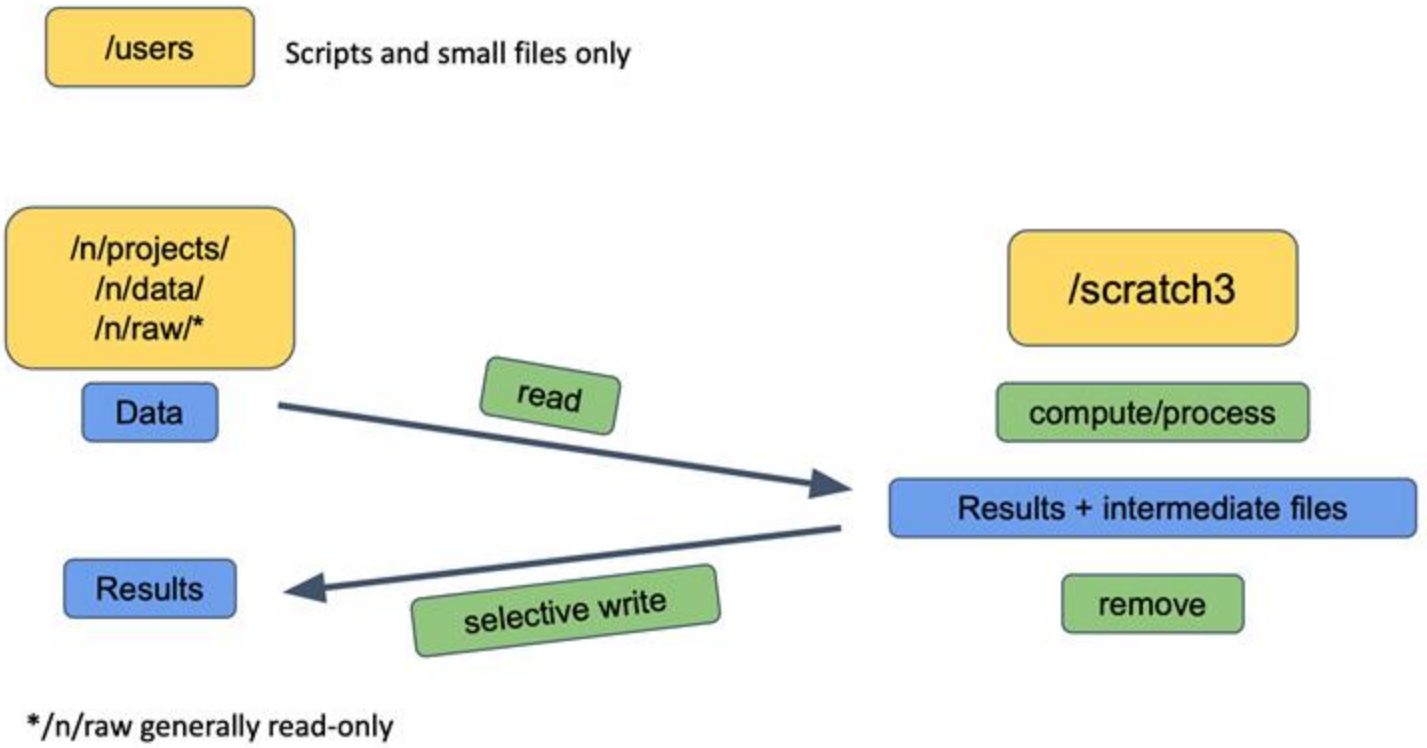
Typical workflow

- At the end of the process, specific data products (e.g. final results) are selected and written into the project directory

- All remaining data removed from the workspace on Scratch

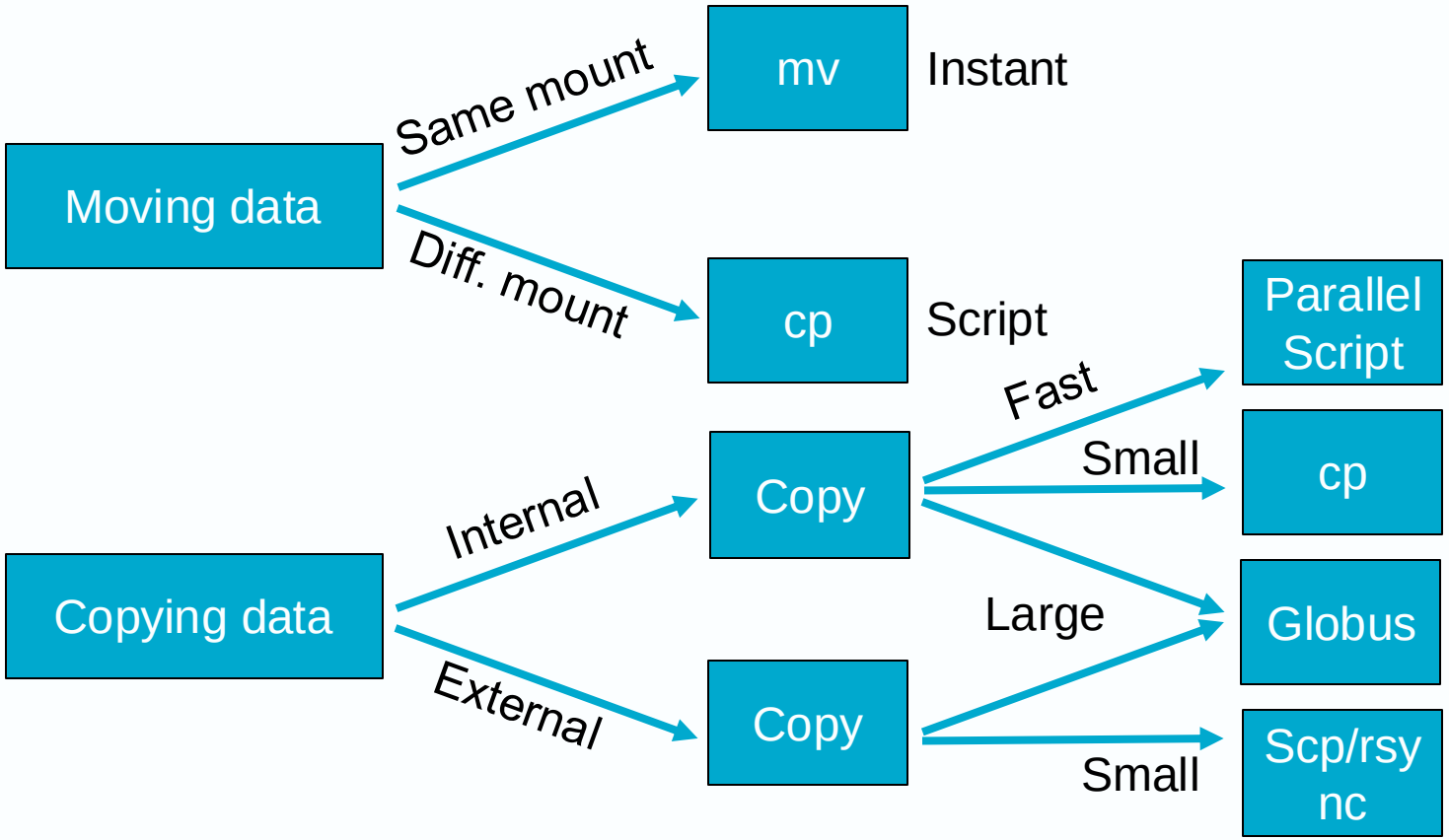
- Two approaches:
  1. Identify products to selectively write (longer-term); remove the rest
  2. Remove what isn't needed; write remainder to longer-term storage
- Typically, many more temporary products than final products
- **First approach** is significantly easier

Typical workflow



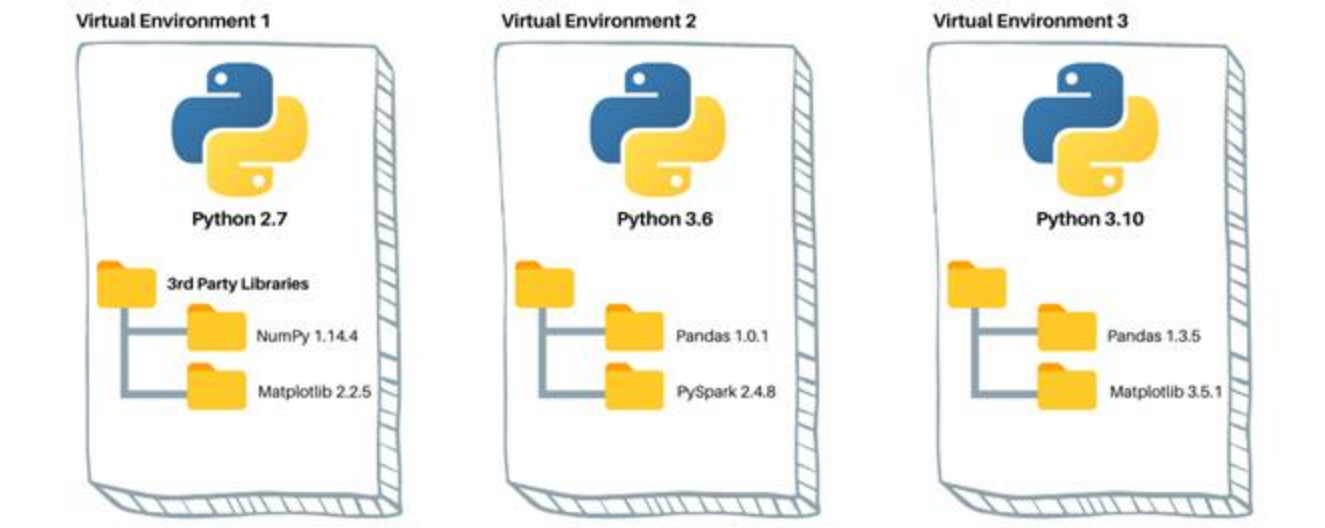
Copying or moving data

- Data can be moved or copied between directories on the same or different mounts / filesystems, or externally



Copying or moving data

- Python virtual environments (venv) should not be moved
  - path associated with environments hard-coded
  - environment can't be activated after changing its location
  - Rather rebuild, and use `pip freeze` to identify the venv packages
  - Generally, recommend building venv in personal workspace
    - e.g. `/idia/users/$USER/software`



Copying or moving data

- Moving within same mount (instant)
  - `mv /idia/users/$USER/run1 /idia/projects/my-project/processed`
- Small copy
  - `cp -a /scratch3/projects/my-project/final-run /cbio/projects/my-project/processed`

Large Transfers with Globus

- Globus recommended for large transfers
- Internal Globus transfers may not perform optimally as **cp**
- But features contained within Globus (“Transfer & Timer Options”) make it useful for internal transfers – e.g.
  - “sync” option to only transfer new or changed files
  - verifying of file integrity
  - schedule regular transfers within particular directories
- Recommend generally enabling “sync” option, and option to “preserve source file modification times”
  - e.g. when having to repair symbolic links (symlinks) with `rsync`

Fast Transfers with Parallel Copy Script

- Fast Transfers
- GNU parallel task recommended for an efficient internal copy on ilifu, to simultaneously transfer many large files
- First, identify directories with many large files/directories
  - The approach performs poorly when run over a small number of files
- e.g. run 16 parallel calls of `cp -a` over 16 files/subdirectories located in /scratch3/users/$USER/my-data directory
- ```
  #!/bin/bash
  shopt -s dotglob #Include hidden files with '*'
  mkdir /ilifu/astro/projects/my-project/my-data
  cd /scratch3/users/$USER/my-data
  printf '%s\n' * | parallel -j 16 cp -a {}
  /ilifu/astro/projects/my-project/my-data
  ```
  ```

Fast Transfers with Parallel Copy Script

- Make it executable and create logs directory
  - `chmod +x parallel_copy.sh`
    `mkdir logs`
- Write sbatch script and run it (`sbatch parallel_copy.sbatch`)!
- ```
  #!/bin/bash
  #SBATCH --cpus-per-task=16
  #SBATCH --mem=16GB
  #SBATCH --job-name=parallel_copy
  #SBATCH --output=logs/%x-%j.out
  #SBATCH --error=logs/%x-%j.err
  #SBATCH --partition=Main
  #SBATCH --time=02:00:00


  export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
  ./parallel_copy.sh
  ```
  ```

Checking File Integrity

- It is important to check the integrity of data before removing it from the original location
- For individual files, this can be achieved using a program such as
  - **md5sum** or **sha1sum – they are fast**
  - There are also some programs **sha256sum**

- For individual files, outputs a checksum
  - Checksum identical for each file if transferred intact

Workflow Maturity

Important to identify maturity of workflow during processing

Stage or type

Affects processing and data management strategy followed

In general, you will be:

1. Prototyping or developing your workflow
2. Running a production workflow

Prototyping

- Involves experimentation
  - Create or optimise workflow
    - Identification of optimal parameters, data products, etc.
- May include
  - Manual processing/ disconnected custom scripts manually run one by one
  - Significant amount of interactivity
  - Running fully automated pipeline
    - Experimenting with different parameters to optimise results

Prototyping

- Intermediate/temporary data products may be retained
  - For comparison, or input to experimentation
- Best practises
  - Experiment with small volumes of temporary products
  - Avoid prototyping / development over large volumes, unless necessary
  - Verify outputs and identify optimal parameters

Production

- Workflows where little to no development, experimentation or interactivity expected to occur
- No interest in retaining intermediate / temporary products
- Generally run as pipeline
  - Contained within series of end-to-end steps
    - i.e. the output of one step is used as input to the next step
  - Often automated and configured before launch
  - Often scheduled to run in advance (e.g. via Slurm and/or Nextflow)
  - Often managed by a single wrapper software package
- Pipeline may be run for prototyping, when workflow or parameters not optimised

Production

- Best practises include:
  - Capture software / pipelines versions and input parameters
    - Results can be reproduced
  - Automate/pipeline removal of temporary data products
  - Automate/pipeline selective write of final products to long-term storage
- Temporary data products can be regenerated at any point via running the same workflow with the same inputs
  - Safe/ideal to remove products automatically during production workflow

General Best Practices

Prototyping should develop into production workflows

Backup your scripts, workflow or pipeline, ideally by uploading to a version-controlled repository such as GitHub

We recommend resources from the Software Carpentry website

For each processing run, keep record of

Software versions of workflow/pipeline

Its input parameters

Products to Retain

- For typical workflow, **final data products** retained for longer-term storage, as produced by your workflow
- Also retain from your workflow
  - Parameters
  - Inputs
  - Versions
  - Logs (e.g. sbatch standard out / error)
- For the purpose of reproducibility and posterity
- More radio astronomy-specific data products to retain later

Products to Remove

- Recommend first selectively writing data products you wish to retain for longer-term storage
- Then remove everything else from processing run
  - Will include temporary / inflated data products
- In some cases, better/easier to identify which products to remove, and write remainder to longer-term storage
- A helpful start is identifying large data products that don't need to be retained and removing those
  - Can run large `rm` within interactive Slurm session, or transfer node
- More radio astronomy-specific data products to removed later

Q & A

Radio Astronomy Data Management

When processing [Multi-] MeasurementSet ([M]MS)

- Important to retain final calibrated data
- A few to 10s TB in size for MeerKAT data
- Ideally with single data column (e.g. corrected data)

Additionally important to retain following data products

- Calibration tables (typically MB in size)
- Flag versions (typically GB in size)
- Final images (typically MB in size) or cubes (typically GB in size)

Possible to store only calibration tables and flag versions

- Remove calibrated data, which can be regenerated at any point
- i.e. apply calibration (applycal) and flags (flagmanager) to raw MS

General Radio Astronomy Workflow

- Typical MeerKAT 32k MS (8-hr, 64 antennas, 4 polarisations) is about 12 TB (18TB)
- Recommended to average or select data wherever possible
  - During SARAO archive transfers (see upcoming demo)
  - Reduces disk volumes and significantly improves data processing time
  - Contact support@ilifu.ac.za to request averaging previous raw data

General Radio Astronomy Workflow

- By default, raw data:
  - Contains single DATA column
  - Stored as read-only MS in /idia/raw (symlinked from /idia/projects)
    - Can be read during initial processing steps (e.g. mstransform / split / partition), ideally on a **scratch** mount
    - For pipelines requiring MS **within the working directory**, or where you wish to give the raw MS a different name recommend creating a **symbolic link**:
      - `cd /scratch3/projects/my-project/processing/`
      - `ln -s /idia/raw/my-project/SCI-YYYYMMDD-PI-01/0123456789/0123456789_sdp_l0.ms my-raw-data.ms`
      - `/scratch3/projects/my-project/processing/my-raw-data.ms` will point to raw read-only MS

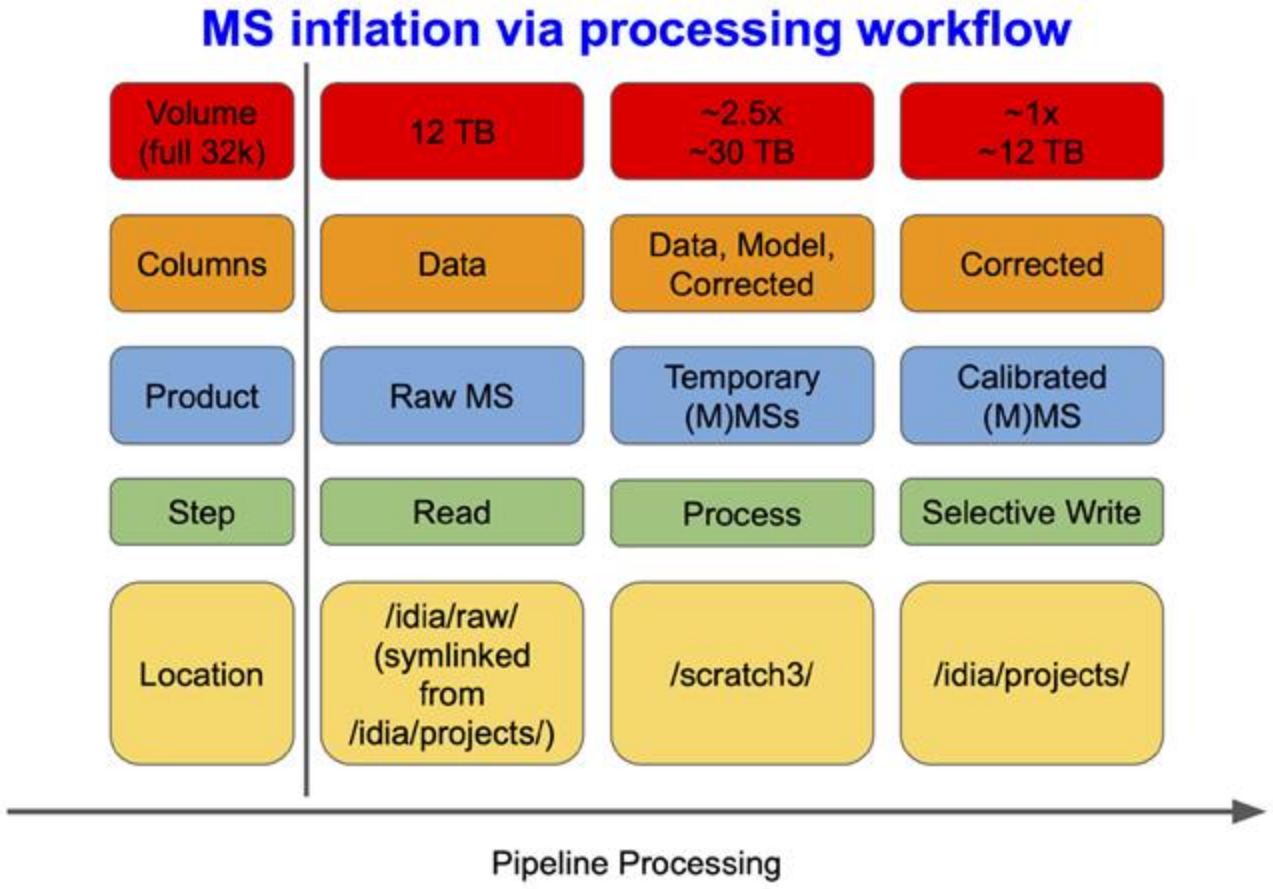
General Radio Astronomy Workflow

- (M)MSs will inflate by ~2.5 times from single DATA column
  - Adds MODEL_DATA (e.g. setjy) & CORRECTED_DATA (e.g. applycal)
  - Often initial cross-calibration will produce these temporary (M)MSs, split out corrected data for target(s), then self-calibrate, further inflating this separate (M)MS with three data columns
- Final calibrated data should contain single corrected data
- Roughly equal to raw data (or smaller with averaging)
  - Selectively written back to project directory for longer-term storage
- All other temporary inflated products should be removed

General Radio Astronomy Workflow

- We also recommend removing raw MS
  - Read-only, so please contact support@ilifu.ac.za to request this
- If retaining raw MS is required for longer verification
  - Please contact support@ilifu.ac.za to motivate for this
- Raw data can be transferred again from the SARAO archive
  - If older than 200 days, first restaged from tape
- May be possible to recover raw data from derived (M)MSs
  - e.g. original DATA column exists; run flagmanager to undo flags
- More info about MeerKAT processing and data management strategies in our MeerKAT processing documentation

General Radio Astronomy Workflow



Find and Remove Measurement Sets

- Expected to remove old files on **scratch** mounts
- As a start, helpful to identify & remove large files not needed
- If you no longer need the (M)MSs from a completed processing run, remove them to free up the bulk of the storage from your processing run.
- **IDIA pipeline**: `cleanup.sh` and `allSPW_cleanup.sh` scripts
  - Remove temporary MMSs, and final calibrated MMSs, respectively

Find and Remove Measurement Sets

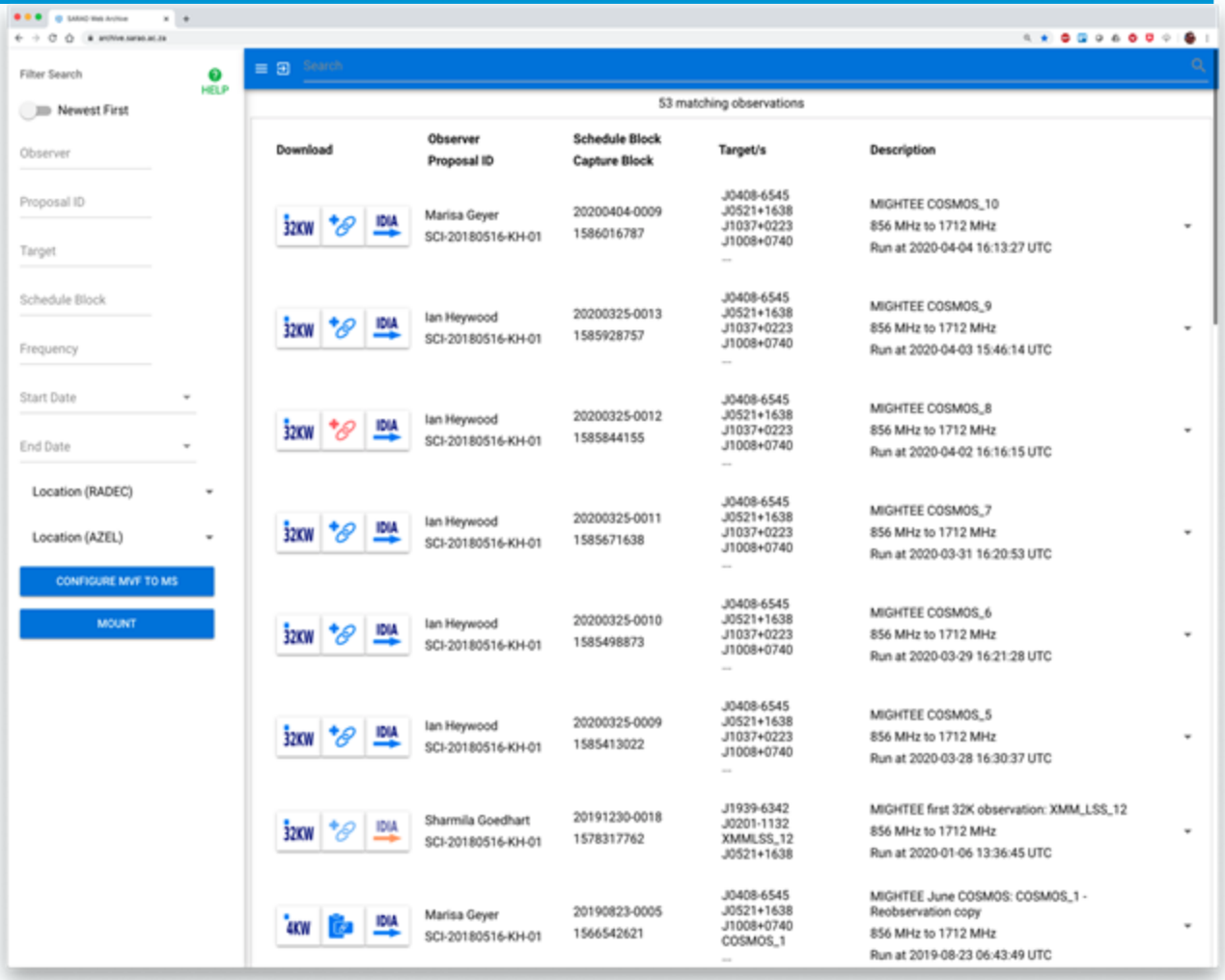
- General workflow: find and remove (M)MSs
  - `#!/bin/bash`

```
find $1 \( -name "*.ms" -o -name "*.mms" -o -name
"*flagversions" \) -exec ls -d {} \; >
vis_and_flags_tmp.txt
du -hsc $(cat vis_and_flags_tmp.txt)
read -p "Press return to remove data... "
rm -r $(cat vis_and_flags_tmp.txt)
rm vis_and_flags_tmp.txt
```

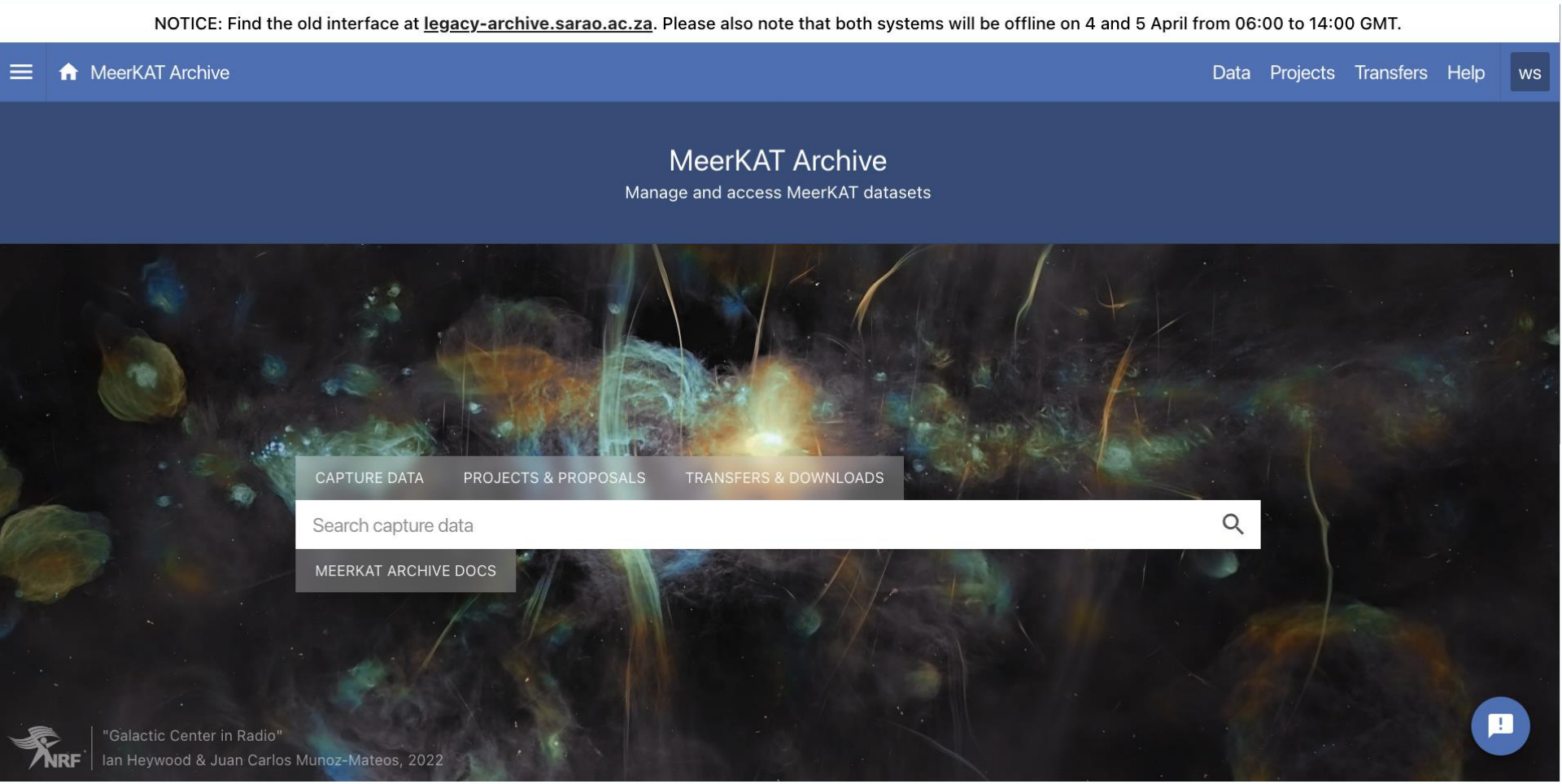
- Finds and displays volume (and total sum) of the (M)MSs
- `read -p` forces return key to remove data, or ctrl+C to cancel
  - e.g. if (M)MS displayed you don't wish to remove

Data Transfers: SARAO archive

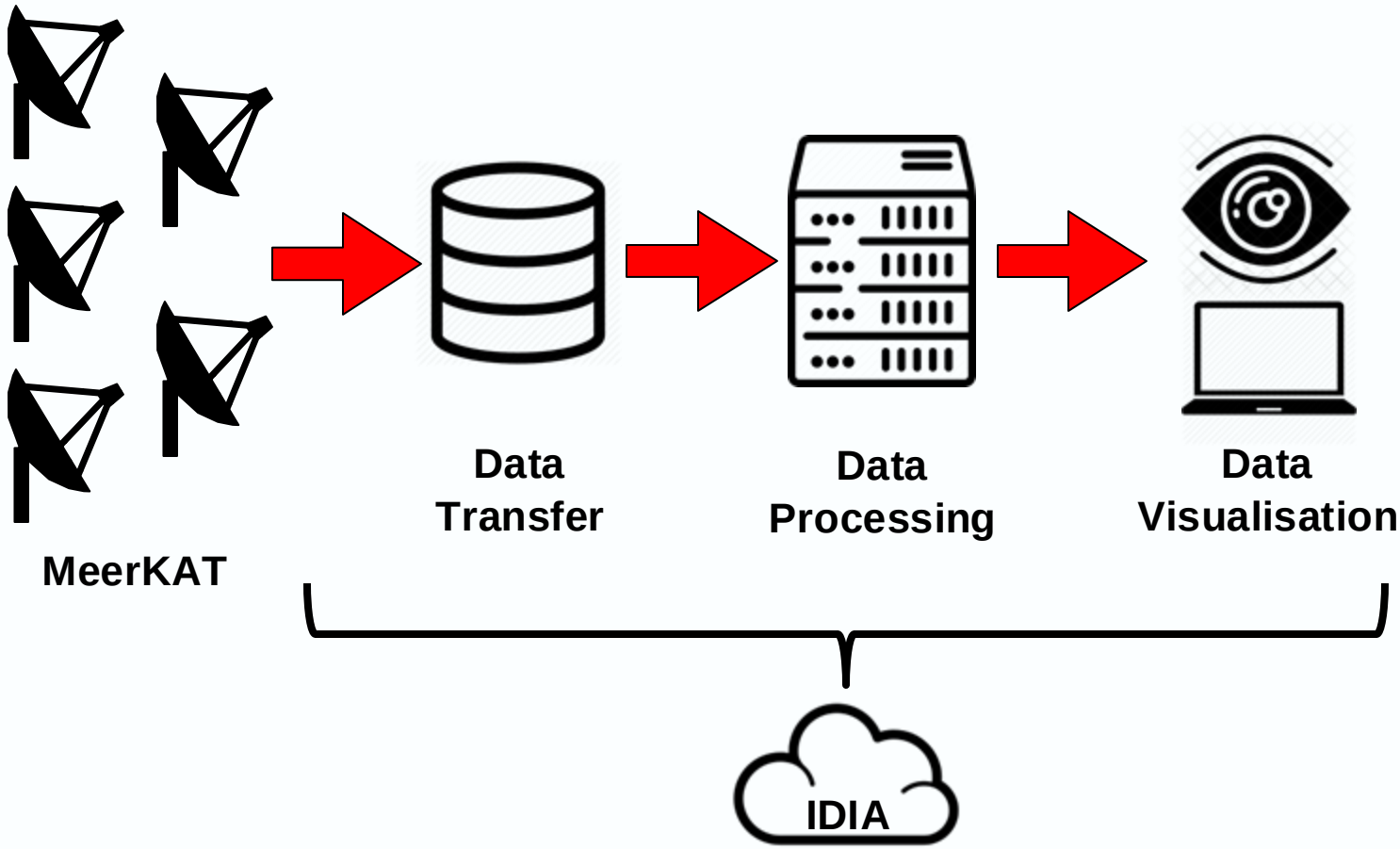
- Documentation
- Can push data to IDIA if given permission by PI
  - First register for archive then PI grant access (or contact service desk)
- Must have an existing ilifu project
  - Contact support@ilifu.ac.za to request



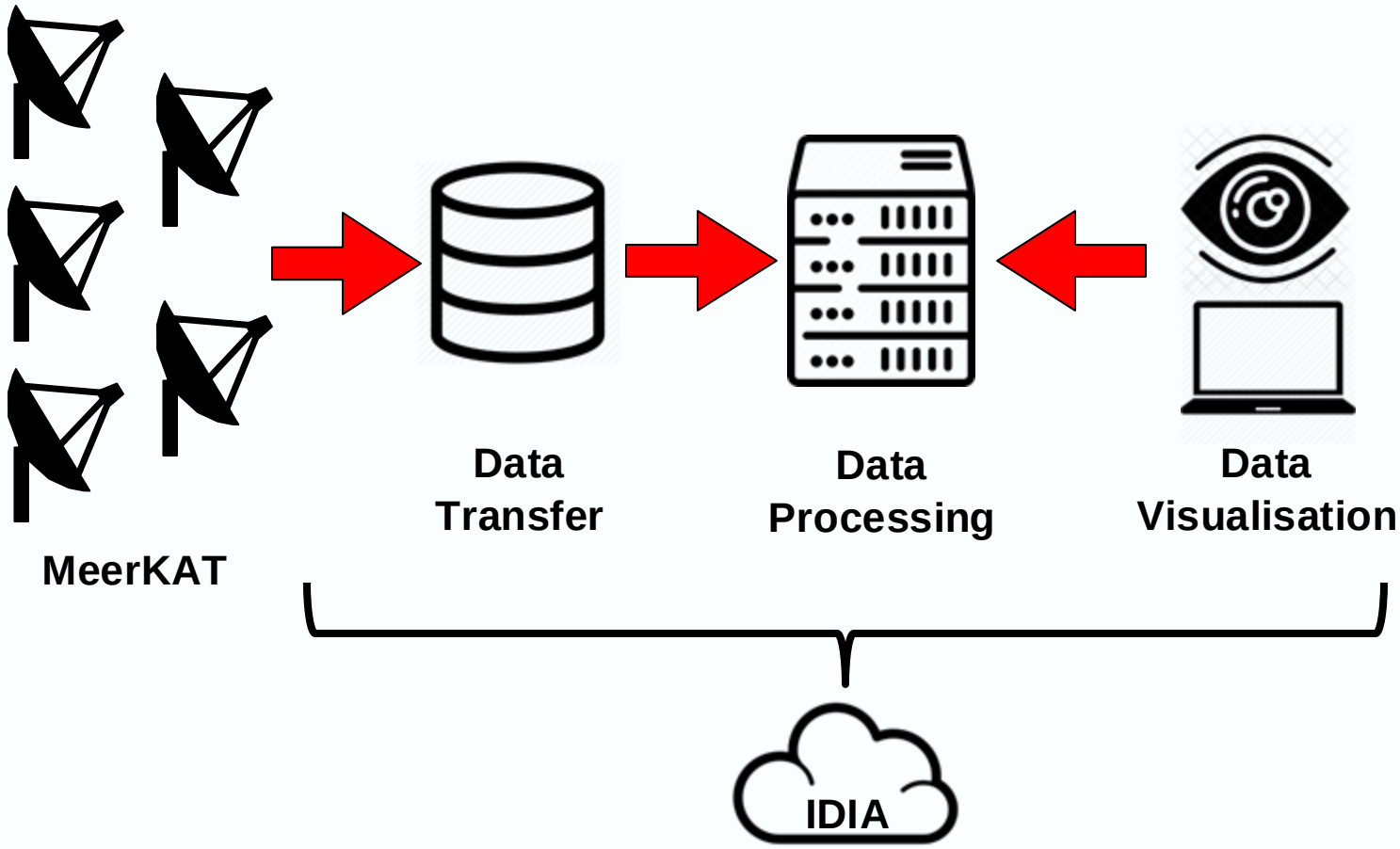
Data Transfers: SARAO archive



IDIA and the MeerKAT toolbelt



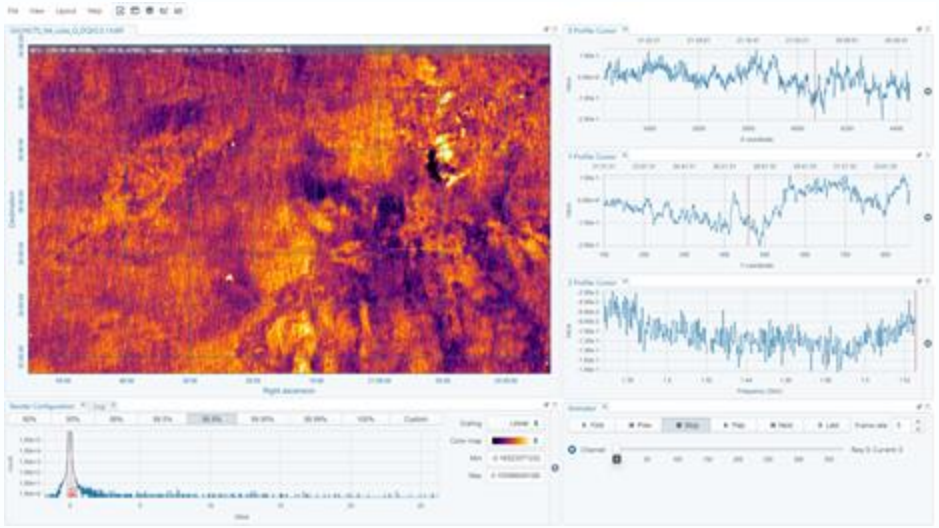
IDIA and the MeerKAT toolbelt



CARTA

- Cube Analysis and Rendering Tool for Astronomy (CARTA)
  - IDIA (South Africa) – NRAO (US) – ASIAA (Taiwan)
  - Cloud-based remote visual analytics of large image cubes
  - Supports many image formats: FITS, CASA, Miriad, and **HDF5**
  - v4.1.0 stable release January 2024 – available here: https://cartavis.org
  - Deployed at ALMA Regional Centres and servers around the world

CARTAvis



IDIA

ilifu

CARTA at IDIA

- docs.ilifu.ac.za/#/astronomy/astronomy_software?id=carta
- CARTA server hosted on ilifu at https://carta.idia.ac.za
  - Login details same as for Jupyter, emailed to you when ilifu account set up
- Beta (development) server: https://carta-beta.idia.ac.za
  - Routinely updated between major releases
- Currently running v5.0.0
  - v5.0.0 release note and feature highlight are available here
- All Astro users (carta-users) have access

CARTA HDF5 format (IDIA schema)

- Strongly recommended for large images/cubes
- Purpose-built HDF5 schema developed at IDIA
  - Efficient FITS to HDF5 converter tool developed at IDIA
- hdf5-converter
- Files go through an “ingestion process”
  - Rotated dataset (XYZ -> ZYX) cached for spectral profile generation
  - Pre-calculated statistics & histogram for image loading (slides & cubes)
  - Pre-calculated tiles for image loading
  - Mipmaps
- Can write IDIA HDF5 file to /carta_share/current/ (SSD) for performance

CARTA HDF5 format (IDIA schema)

- HDF5 converter usage:
  - `srun fits2idia -o {OUTPUT HDF5 file} {INPUT FITS file}`
- Suggested to copy straight into /carta_share subdirectory:
  - `srun fits2idia -o /carta_share/users/$USER/image.hdf5 image.fits`
- Speed-up achieved by increasing CPUs & memory allocation:
  - Up to 251 GB for node in Main partition, and 1508 GB HighMem partition
  - `srun --mem=50GB --time=5 --cpus-per-task=10 fits2idia -p -o /carta_share/users/${USER}/image.hdf5 image.fits`
  - -p shows simple progress bar

CARTA HDF5 format (IDIA schema)

- Some large FITS cubes will exceed max memory values
  - Unable to convert to HDF5 in default mode
- Option -m will report predicted memory usage and exit:
  - `fits2idia -m image.fits`
- If exceeds 1508 GB (or 251 GB if HighMem nodes unavailable):
  - Option -s must be used, using slower but less memory-intensive method
  - Single CPU iterating through a single channel at a time
  - `srun --mem=10GB --time=01:00:00 --cpus-per-task=1 fits2idia -s -p -o /carta_share/users/${USER}/image.hdf5 image.fits`
- Predicted memory usage for slow-mode conversion reported when using both options -s and -m

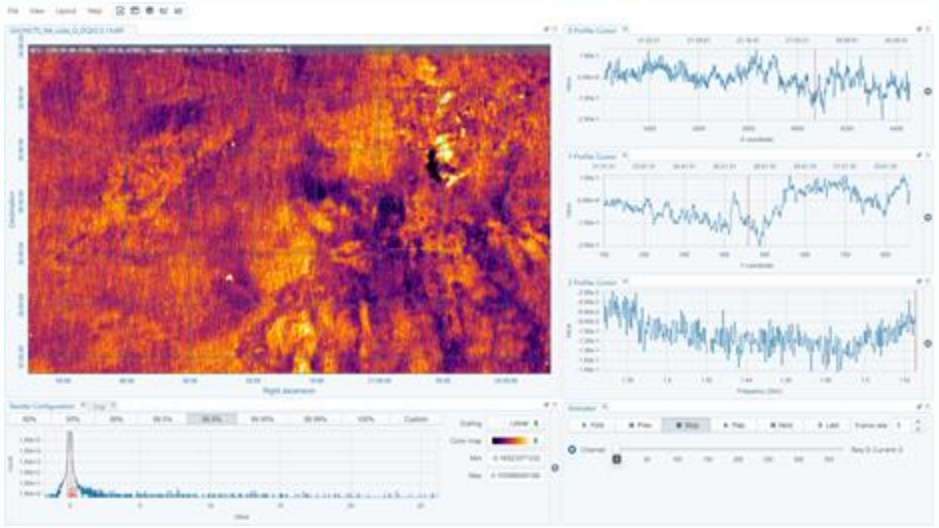
CARTA at IDIA

- By default, CARTA will browse in /carta_share/users/$USER
- Can access any files or folders in /carta_share, /scratch3, /idia and /ilifu directories that your ilifu user can access
  - /idia and /ilifu mounts read-only, so cannot export files here
    - e.g. region files, FITS cutouts or moment maps
    - Can export to /carta_share or /scratch3
- Many copies of images in /carta_share (HDF5, FITS, CASA or miriad) no longer necessary, unless for performance reasons
  - For large cubes/images, we recommend converting to HDF5 and writing to /carta_share/current (SSDs)
    - Superior performance during visualisation
  - Not necessary for all other images, as visualising directly from /idia, /ilifu or /scratch3 in any format will still result in good performance

CARTA at IDIA

- Can move the files/folders to relevant project directory, within /carta_share/groups/, /scratch3/projects/, or /idia/projects/
  - e.g. for other project members to access specific files
  - Please request support to create relevant project directory if it doesn't exist

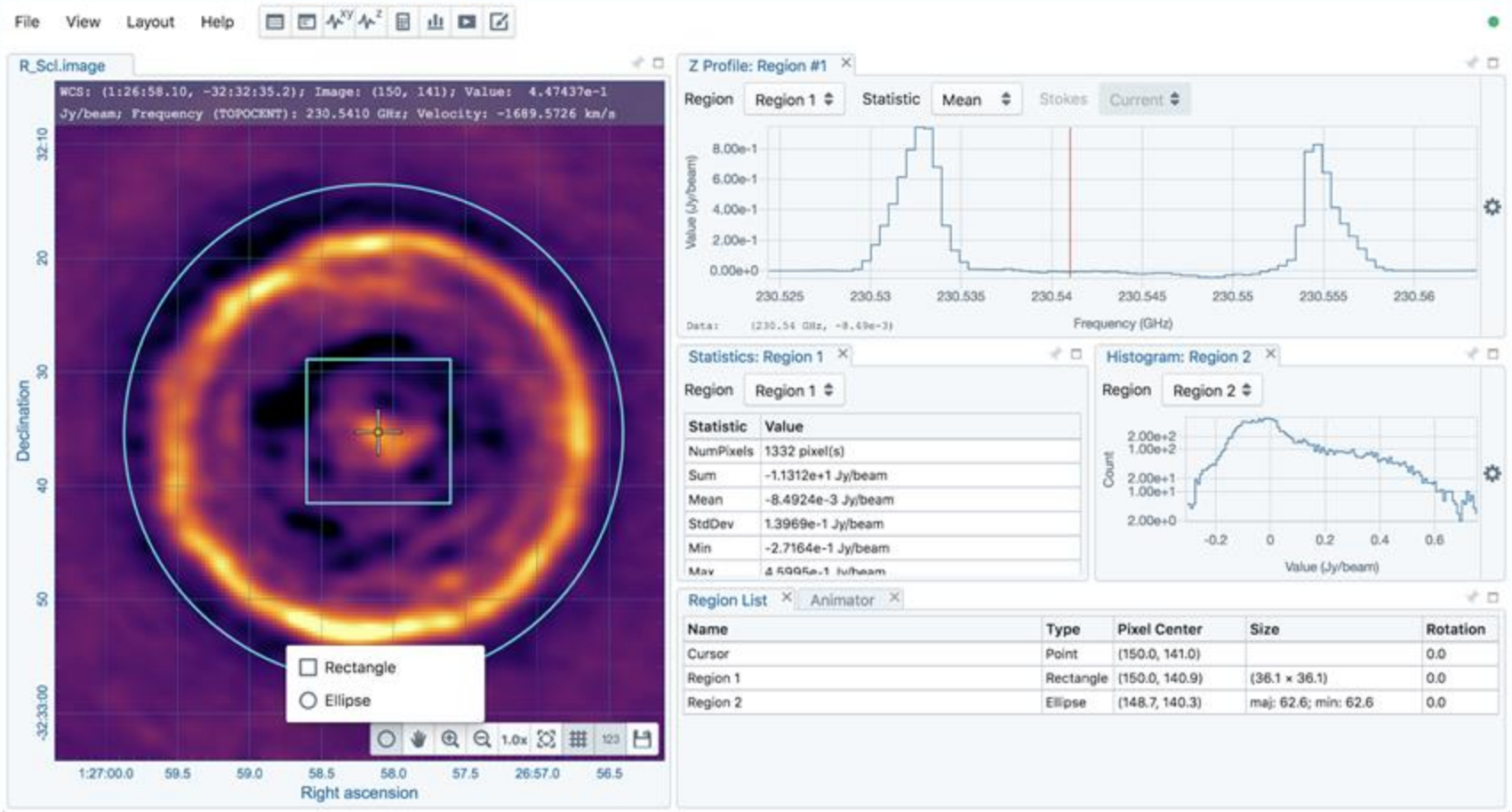
CARTAvis



CARTA at IDIA: troubleshooting

- Please contact support@ilifu.ac.za if you don't have access
- For CARTA-specific issues
  - Contact CARTA helpdesk or file GitHub issue
- Restarting your backend
  - e.g. issues starting CARTA, or CARTA session crashes
  - File -> Server -> Restart Service or visit carta.idia.ac.za/dashboard
    - Also accessible via File -> Server -> Dashboard
  - Press the button to "Restart CARTA service"
  - Refresh your CARTA page or press the button for a "New CARTA session"
- Hard-reload white screen
  - e.g. white screen appears on IDIA CARTA server
  - Force reload (e.g. command-shift-R) page to clear cache, often a few times

CARTA demo


File
View
Layout
Help
R_Scl.image
WCS: (1:26:58.10, -32:32:35.2); Image: (150, 141); Value: 4.47437e-1
Jy/beam; Frequency (TOPOCENT): 230.5410 GHz; Velocity: -1689.5726 km/s
Declination
Right ascension
Rectangle
Ellipse
Z Profile: Region #1
Region
Region 1
Statistic
Mean
Stokes
Current
Value (Jy/beam)
Frequency (GHz)
Statistics: Region 1
Region
Region 1
Statistic
Value
NumPixels
1332 pixel(s)
Sum
-1.1312e+1 Jy/beam
Mean
-8.4924e-3 Jy/beam
StdDev
1.3969e-1 Jy/beam
Min
-2.7164e-1 Jy/beam
Histogram: Region 2
Region
Region 2
Count
Value (Jy/beam)
Region List
Animator
Name
Type
Pixel Center
Size
Rotation
Cursor
Point
(150.0, 141.0)
0.0
Region 1
Rectangle
(150.0, 140.9)
(36.1 × 36.1)
0.0
Region 2
Ellipse
(148.7, 140.3)
maj: 62.6; min: 62.6
0.0

Thank You