

Pawsey Guide for MWA(sci) folks

Containers on Garrawarla (and elsewhere):

The Garrawarla cluster doesn't mount /group so you'll have to ensure that your code is in /astro if you want to access it. Since most of our software was in /group we have started porting the software to singularity containers in /pawsey/mwa, which is mounted on Garrawarla.

To use containers:

- Load the singularity module using
 - `module load singularity`
- Choose the container that you want to run:
 - They are stored in `/pawsey/mwa/singularity/<container_name>`
 - See `/pawsey/mwa/singularity/README.md` for an overview of which containers contain which software
- Run the container via (example)
 - `singularity exec -B $PWD /pawsey/mwa/singularity/wsclean/wsclean_2.9.2.img wsclean`
- See the [Pawsey support page](#) for more info about singularity
- Contact one of the container maintainers if you have questions about a specific container (See the README.md file)

Containers and their contents for people who are migrating away from MWA_tools module:

The following containers are likely to be useful to MWASCI members as a replacement for the MWA_Tools module that used to work on Magnus/Zeus. MWA_Tools module doesn't work any more because of changes that pawsey have made to the available modules and versions (mostly python issues). The MWA_Tools module will not be resurrected.

Container image	software within container
<code>/pawsey/mwa/singularity/casa/casa.img</code>	<code>casa casaviewer</code>
<code>/pawsey/mwa/singularity/cotter/cotter_latest.sif</code>	<code>aoflagger aoqplot aoquality cotter fixmwams rfigui</code>
<code>/pawsey/mwa/singularity/wsclean/wsclean.img</code>	<code>wsclean aoflagger casacore chgcentre python2.7 python3.6 taql</code>
<code>/pawsey/mwa/singularity/mwa-reduce/mwa-reduce.note-1.img</code>	<code>adding adding aegean2model apparently applybeam applyion appliesolutions autoprocess bbs2model beam calibrate cluster editmodel fitsmodel flagantennae flagbaselines flagionsolutions flagmwa flagsolutions flagsubbands ionpeel matchsources mrc2model mwafinfo pbadding pbcorrect peel phasecal regridding render scaleimage sedcombine solutiontool storetime subtrmodel vo2model</code>
<code>/pawsey/mwa/singularity/python/python_latest.sif</code>	stilts python3 numpy cython scipy matplotlib astropy aiipy h5py healpy ipython pandas pyyaml tqdm colorama patsy statsmodels AegeanTools mwa-hyperbeam pyuvdata PyBDSF
<code>/pawsey/mwa/singularity/rfi_seeker/rfi_seeker.img</code>	RFISseeker python3 astropy numpy scipy matplotlib wcstools
<code>/pawsey/mwa/singularity/robbie/robbie-next.sif</code>	Robbie-next topcat stilts python3 AegeanTools astropy healpy numpy scipy lmfit pandas matplotlib
<code>/pawsey/mwa/singularity/manta-ray-client/manta-ray-client_latest.sif</code>	MWA client

Notes:

1. mwa-reduce relies on the mwprimary beam model from the "mwapy" package, which is NOT included in the singularity container because it's rather large. To invoke this container and have it work properly you should use:
 - a. `>singularity exec -B /pawsey/mwa:/usr/lib/python3/dist-packages/mwapy/data /pawsey/mwa/singularity/mwa-reduce/mwa-reduce.img <command>`

Setup:

- Remove the `module load` and `export` commands from your `.bashrc` and `.profile` files

- See [README](#) on the manta-ray client github for how to setup asvo access via a batch script.
- Contact your admin rep (Paul/Chris/Sammy) before contacting help@pawsey as there are some things that we can fix faster.
- Only members of the group `mwaadmin` can build software and edit the module files in `/group/mwa/software`. We have automated the build process for future reference. If you require software that is not in there, and which is to be used by more than 1-2 people, then contact one your admin rep and we will work something out.

Best practice:

- Don't rely on `.profile` or `.bashrc` to load the modules that you need within a batch script
- Use containers as described above.
- To ensure a 'clean' environment at the start of your batch script you should use:
 - `source /group/mwa/software/module_reset.sh` # to reset the modules to some default/minimal list. NOTE: you must source this script, don't run it directly.
- At the start of each job script load the modules that you need using:
 - `module use /group/mwa/software/modulefiles`
 - `module load MWA_Tools/mwa_sci` # will give you access to the MWA software stack.
 - `module load python` # will give you python/astropy/numpy/scipy (included in MWA_Tools)
 - `module load stilts` # will give you java and stilts (included in MWA_Tools)
- Data use:
 - `/group` and `/astro` should be used for storing code and data. Intermediate/long term data products should be on `/group` and `/astro` should only be used for temp/short term data as it's not backed up.
 - `/home/` or `~` should be used only for config files like `.profile` `.bash_aliases` etc..
 - DO NOT store scripts, logfiles, or code in your home directory.
 - if you make symlinks in your home directory for convenience, don't use these links in your job scripts, it will put a strain on this file system for all users.
 - downloading and moving of data should be done via the Zeus copyq (for data <1GB you may get away with the occasional `wget/scp`)
 - copying data FROM pawsey should be done via hpc-data.pawsey.org.au

How to check your group quota:

If the `mwa{sci,eor,ops}` quota on `/astro` or `/group` is exceeded then all users in the given group will suddenly be unable to write files to the given device. This will cause jobs to fail in unexpected ways. To get a quick view of the current usage for your default group you should use:

```
pawseyAccountBalance --storage [-project=<group>]
```

Cross cluster jobs:

It is possible to submit jobs to the Zeus copy queue whilst logged into Galaxy. Just use `sbatch -M Zeus -p copyq <script.sh>` from the command line.

It is important that your job script has the following two lines in the header or cross-cluster job submission will give you some strange errors:

```
#!/bin/bash -l
#SBATCH --export=NONE
```

Moving data:

Here is a script for copying data from `/group` to `/astro`.

```
#!/bin/bash -l

#SBATCH --export=NONE
#SBATCH --account=mwasci
#SBATCH --clusters=zeus
#SBATCH --time=12:00:00
#SBATCH --nodes=1
#SBATCH --tasks=4
#SBATCH --cpus-per-task=1
#SBATCH --partition=copyq

module load mpifileutils

# Edit this if your files are kept elsewhere
mydir=/group/mwasci/$USER

# Edit this to change which directory gets copied
dirtocopy=projectstuff

# Edit this if you want to put your files somewhere else
destdir=/astro/mwasci/$USER/destination

if [[ ! -d $destdir ]]
then
    mkdir $destdir
fi

# Note that in order to use four tasks, you have to explicitly use --tasks=4 in the SBATCH header.
mpirun -np 4 dcp -p $mydir/$dirtocopy $destdir/$dirtocopy
```

Put those instructions in a script (e.g. copyscript.sh) and then run (e.g.) sbatch -M zeus copyscript.sh .

Note that this will work in the copyq for up to 48 hours. I have copied about 20TB of data in less than 24h, so this should be suitable for most purposes. Note that this does NOT remove the old data! See the next entry to the guide.

Also note time limit may default to something short if you do not include it.

Deleting data:

From: <https://support.pawsey.org.au/documentation/pages/viewpage.action?pageId=29263957>

Using the standard Linux command rm to delete multiple files on a Lustre filesystem is not recommended. The rm command will generate a stat() operation for each file it removes, meaning all of the attributes of a file (filetype, owner, permission, modification time, etc.) will be returned. A large number of stat() operations can place an increased load on the metadata server, resulting in instabilities with the filesystem. Instead, users should use munlink, a Lustre-specific command that will simply delete a file without performing a stat() operation. Below is an example of how to remove files using munlink:

```
find ./processor0 -type f -print0 | xargs -0 munlink
```

Here's an overview of each step in that command:

```
find ./processor0 -type f
```

The find command will search the directory processor0 and all subdirectories for anything that is a file (-type f) as opposed to directories (-type d). -print0

This dictates how the returned list of files is formatted, and ensures that they are in the correct format for the next step (xargs).

```
| xargs -0 munlink
```

The list of files is passed to the xargs command using the pipe, or |, command. xargs will then convert the list of files, line by line, into an argument for whatever command is specified at the end (in our case munlink). The -0 flag is related to the format of the listed files; if you use -print0 in the find command you must use -0 in the xargs command. Once all of the files are deleted you can remove directories with a similar command:

```
find ./processor0 -depth -type d -empty -delete
```

Again, the find command will search the directory processor0 and all subdirectories for any empty directories and delete them. The depth flag instructs finddepth Process each directory's contents before the directory itself. The -delete action also implies -depth. to process a directories contents before the directory itself. Please note that the flags passed to find are evaluated as an expression, so if you pass -delete first, find will attempt to delete below your starting directory.

Using the HSM:

You should get a confirmation email if your allocation is approved. By default you'll get 10TB, but you can argue for more if you have a convincing case.

See <https://support.pawsey.org.au/documentation/display/US/Working+with+the+HSM> for details on how to copy data to and retrieve data from the long-term storage.

A short version is:

- this is a long-term archive for finished data products, which will be put on tape;
- putting them in the archive is as simple as ssh-ing into [hpc-hsm.pawsey.org.au](https://support.pawsey.org.au), and copying or using dcp to copy from /group or /astro to /project /\$project (the project number you get allocated)
- every six hours, the archive system makes a list of files in that directory and moves them to tape;

- they must then be 'recalled' if you want to access them again, so you must be careful not to treat files on /project/\$project as if they are on a 'normal' spinning-disk filesystem.

The documentation has all of the details about the tools to use to check the status of your files. PLEASE read this very carefully before logging in and doing anything.

Since this is a tape system, I highly recommend tarring directories like measurement sets, since otherwise you will have to recall zillions of nested files, which will be messy (and not good for the robot). A good way of doing this is to use the workq and the command

```
module load pigz
```

```
tar -I pigz -cf sometarball.tar.gz somedirectory.ms/
```

pigz will do a multi-threaded zip which is faster than single-thread.