ISTA ASTRO SOFTWARE

September 2024

Contents

Introduction

Last edited by: Aayush Desai

Hello world. This is the ISTA Software Page, which will hopefully have all the neat tools and tricks the department would have developed over the years. The repository is maintained by Aayush Desai for the time being, and is open to contributions from anyone who would like to add to it.
Here you can find access to example scripts that are group agnostic (mostly). Should you have any ideas for further developments, just add another section and start writing.

For those working on Overleaf, once you are done with the edits, please make sure to push the changes to the repository. You can do this by:

1. **Committing the changes:** This can be done by clicking on the "menu" button, on the top left corner of the screen, navigating to "github" and selecting commit changes. This will save the changes you have made to the repository.
2. **Pushing the changes:** This can be done by clicking on the three dots on the top right corner of the screen, and selecting push changes. This will push the changes you have made to the repository.
3. **Syncing the changes:** This can be done by clicking on the three dots on the top right corner of the screen, and selecting sync changes. This will sync the changes you have made to the repository.

For those working on the local machine, please make sure to:

1. **Pull from the main repo** git pull from the repository to get the latest changes.
2. **Make changed** make the changes you need to make.
3. **Push to the main repo** git add, git commit, and git push the changes to the repository.

1 Sharing Repositories via GIT

Last edited by: Aayush Desai

2 MESA on Cluster

Last edited by: Lukas Einramhof

IT has installed `MESA` on the cluster. In order to get `MESA` running on the cluster, a few steps have to be done to ensure `MESA` is running correctly.

Two templates `slurm` scripts (one for a normal job and one for an array job) are available in this repository under `Templates` (`run_mesa.sh` and `run_mesa_array.sh`).

First the (current version of the) `MESA` module has to be loaded via

```
module load mesastar/23.05.1
```

where '23.05.1' is one of the versions of `MESA` installed on the cluster. If other versions are needed you need to write an email to IT with the specific version that they should install.

To get a template directory for your `MESA` run (if you don't have one yourself) you can copy the `$MESA_DIR/star/work/` directory into your cluster directory. Note that the `$MESA_DIR` command is only available after loading the `MESA` module.

One big caveat with `MESA` installed on the cluster is, that the default cache folders in the `$MESA_DIR` can't be written to. Thus, for every run you have to define a custom cache folder in your personal cluster folder. Furthermore, each `MESA` run has to have its own cache folder. This is especially important if you run array jobs. To set up a custom cache you will have to run

```
mkdir path_to_custom_cache_dir
export MESA_CACHES_DIR=path_to_custom_cache_dir
```

The first command creates a custom cache directory at some specified path, and the second command then sets the MESA_CACHES_DIR flag to the corresponding directory which tells MESA where to put the cache files. Since each MESA run needs its own cache directory, I would suggest putting the caches folder directly into the directory where the specific MESA run is set up. For example, create a folder structure like

```
work/
|
+-- make/
+-- src/
+-- caches/
+-- clean
+-- mk
+-- rn
+-- ...
```

and then add the command

```
export MESA_CACHES_DIR=work_dir/caches
```

into your slurm script.

With all this setup, there should be no issues in running MESA on the cluster.

3 Tips for working on the Cluster

Last edited by: Lukas Einramhof

Cluster status here

3.1 Setting up your environment/Transitioning from Anaconda to Miniforge3

Last edited by: Ivan Kramerenko

Owing to the recent decision of Anaconda to charge a licence fee per user to institutions, the Astro department has decided to transition from Anaconda to Miniforge3. Miniforge3 is a minimal installer for conda, which is a package manager that can be used to install Python packages. The following steps can be used to transition from Anaconda to Miniforge3.

1. Uninstall Anaconda

   (a) (Optional) Backup your conda environments – run the following commands for each environment you want to create a backup for (e.g., some useful environment with many packages installed).

   ```
   conda activate environment-name
   conda env export > environment-name.yml
   ```

   (b) (Optional) Backup your '.condarc' file if you have it in your home directory.

   ```
   cp ~/.condarc ~/.condarc.bkp
   ```

   (c) Remove conda initialization scripts from the terminal shell profiles.

   ```
   conda activate
   conda init --reverse --all
   ```

   (d) Remove the Anaconda directory.

   ```
   rm -rf ~/anaconda3
   ```

   (e) If you're using '**miniconda**' instead, remove the '**miniconda3**' directory.

```
rm -rf ~/miniconda
```

(f) Remove the hidden '**.conda**' file and the '**.conda**' directory.

```
rm -rf ~/.condarc ~/.conda
```

(g) Restart the terminal

2. 2. Download and install Miniforge3 (https://github.com/conda-forge/miniforge)

   (a) Download the installation script.

   ```
   curl -L -O "https://github.com/conda-forge/miniforge/
   releases/latest/download/Miniforge3-$(uname)-$(uname -m).sh"
   ```

   (b) Run the installation script (run the command as is). Answer **yes** when asked, "Do you wish to update your shell profile to initialize conda automatically?"

   ```
   bash Miniforge3-$(uname)-$(uname -m).sh
   ```

   (c) Restart the terminal

   (d) (Optional) Recreate your previous conda environments.

   ```
   conda env create -f environment-name.yml
   ```

3. Use the standard '**conda create**' and '**conda install**' commands to create new environments and install Python packages. Note that '**conda-forge**' will be the new default channel used for package installation.

3.2 File Servers and Structuers

Last edited by: Aayush Desai

Please check the IT Documentation for information on file servers and structures. This will help you understand where to store your data, and how to access it from different locations.

Briefly speaking there are 3(.5) main types of storage:

1. Cluster Storage
2. Archival Storage
3. Standard Storage
4. (Optional) Cloud Storage

```
Note:  Storage on all three systems is shared between the group members, and not equipartitioned.
This means that if one member uses all the storage, the other members will not be able to use
it.  Please make sure to use the storage judiciously.
```

3.2.1 Cluster Storage

What is it?
Whenever you log in to any of the headnode via `<username>@headnode.hpc.ista.ac.at`, your immediate directory is part of the cluster storage. By default your account is under your supervisor. By default, each group has **5TB** of storage, equally accessible by all group members.
The data here is not backed up at all and is thus vulnerable to loss. Please make sure to back up your data to the archival storage or standard storage.

Whats the point?
Since compute jobs are run on the cluster,

1. It is easier to access the data from the cluster storage.

2. The special design of the system allows for quick ($\approx$ 10Gbps) data transfer between the cluster storage and the compute nodes. Thus it is ideal for data that needs to be processed/analysed.

Where is it?
The cluster storage is located at `<username>@headnode.hpc.ista.ac`.

Example Data

1. Light Curves.
2. Full Frame images (yet to be analysed).
3. Simulation/evolution models.

3.2.2 Archival Storage

What is it?
The archival storage is a backup storage system that is used to store data that is important and needs to be backed up or data that has been analysed and no longer needs to be accessible quickly.
The data here is backed up and thus safe from loss.

Whats the point?

1. The data is backed up and thus safe from loss.
2. The data is not accessible quickly, and thus is ideal for data that is not needed for immediate analysis.
3. Frees up space on the Cluster Storage.
4. The data can be accessed from anywhere on the IST (including VPN access).
5. The data can be accessed from the cluster.

Where is it?
One can access the archival storage in two ways:

1. Via the cluster:
   (a) Login to the headnode: `<username>@headnode.hpc.ista.ac`
   (b) The archival storage is located at `/drives/archive/group (or /archive3/group/)`
2. Mapping the drives on your local machine

Example Data

1. Everything on the cluster storage.
2. Data/Analysis that has been published.
3. Raw fits files.

3.2.3 Standard Storage

What is it?
Consider it like ISTAs version of google drive, but for your group i.e. everything there can be accessed by everyone in that group.
The data is backed up daily and available for 1 year.

Whats the point?

1. Points 1-4 from Archival Storage.
2. Accessible to the entire group.
3. If running low on-device storage, a good option to store personal files temporarily.

4. An additional advantage is that if you are using an ISTA-issued machine, all data saved on it locally is available via the Standard storage (and vice versa) on other devices.

Where is it?

1. Via the cluster:
   (a) Login to the headnode: `<username>@headnode.hpc.ista.ac`
   (b) The archival storage is located at `/drives/home (or /fs3/home/)`
2. Mapping the drives on your local machine

Example Data

1. Personal files (temporarily).
2. Group files.
3. Analysis files from the cluster (not raw data)
   (a) This is particularly helpful during Cluster maintenance, as you can still access your data.
   (b) Via the mapping ability, you can access the data from your local machine.

3.2.4 Cloud Storage

Seafile is a cloud storage system that is available to all ISTA members. It is a secure and easy way to store and share files. The data is backed up and can be accessed from anywhere. For more information on Seafile, please refer to the IST documentation.

3.3 Making your setup more efficient to get time on the cluster faster

Last edited by: Lukas Einramhof

After you have run a job for the first time you can check how much CPU and memory it used to then tailor your next run to use just enough resources. This will let your next job run earlier since it uses less resources. To do that run

```
    module load seff
    seff JOB_ID
```

This will give you output similar to

```
    ...
    Job ID: 23693144
    Array Job ID: 23693144_9
    Cluster: istscicomp
    User/Group: leinramh/bugnegrp
    State: COMPLETED (exit code 0)
    Nodes: 1
    Cores per node: 4
    CPU Utilized: 12:13:44
--> CPU Efficiency: 95.75% of 12:46:16 core-walltime
    Job Wall-clock time: 03:11:34
    Memory Utilized: 15.75 GB
--> Memory Efficiency: 49.21% of 32.00 GB
```

With the `CPU Efficiency` and the `Memory Efficiency` you can adjust how much memory and CPUs you ask for in the next run, to get faster priority (unless you need more resources of course). In the example above I could have asked for half the amount of memory.

3.4 Running on the head node vs. submitting via slurm scripts

Last edited by: Aayush Desai

As a general rule of thumb, it is always better to submit jobs via slurm scripts, rather than running them on the head node. This is because the head node is a shared resource, and running jobs on it can slow down the system for everyone else. Furthermore, running jobs on the head node can lead to the job being killed if it uses too many resources or whenever you logout.

There are really helpful slurm scripts on the IT page, which can be found at. These scripts can be used to submit jobs to the cluster, and can be modified to suit your needs.

Jobs that could be run on the head node:

1. Small jobs that require less than 5 minutes to run.
2. Jobs that require less than 1GB of memory.
3. Jobs that require less than 1 core.
4. File IO operations: eg copying files, moving files, etc[1].

From personal experience (as of October 2024), I prefer to use the `eta293` head node as my default login node (found at <`username`>@eta293.hpc.ista.ac.at). It is the largest head-node (300 cpu cores and 1TB of memory).

3.5 Using salloc to get an interactive session

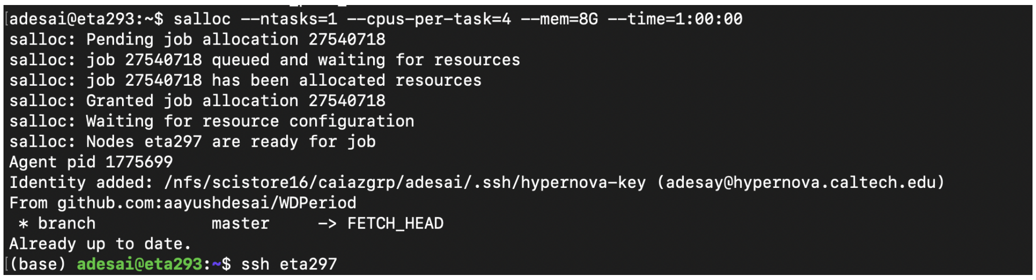
Last edited by: Aayush Desai

Should you find yourself testing some code and needing to run it on the cluster, you can use the `salloc` command to get an interactive session on the cluster. This is particularly useful when you need to test code that requires more resources than the head node can (or should) provide (see 3.4).

`salloc` performs the same way a slurm script would perform.

1. You define the resources you need in a single line
2. It allocates resources for you.
3. You ssh into the allocated node.
4. You run the job.
5. Iterate

This is more convenient than reserving a node as only the **PI can do the latter**.



```
adesai@eta293:~$ salloc --ntasks=1 --cpus-per-task=4 --mem=8G --time=1:00:00
salloc: Pending job allocation 27540718
salloc: job 27540718 queued and waiting for resources
salloc: job 27540718 has been allocated resources
salloc: Granted job allocation 27540718
salloc: Waiting for resource configuration
salloc: Nodes eta297 are ready for job
Agent pid 1775699
Identity added: /nfs/scistore16/caiazgrp/adesai/.ssh/hypernova-key (adesay@hypernova.caltech.edu)
From github.com:aayushdesai/WDPeriod
 * branch            master     -> FETCH_HEAD
Already up to date.
(base) adesai@eta293:~$ ssh eta297
```

Figure 1: Example of salloc command

```
salloc --ntasks=1 --cpus-per-task=4 --mem=8G --time=1:00:00
```

If you are done early, please remember to `scancel` your job, else the job run for the entire duration of the entire time.

[1]If you are running such a job on the head node, and you would like to move it to the cluster, you can use the `screen` command. This command allows you to run a job in the background, and can be detached from the terminal. This way, you can run the job on the head node, and then detach it and move it to the cluster. Please do not misuse this to run bigger jobs on the headnode. Lets try to respect the *shared* resource

4 Working on GIT

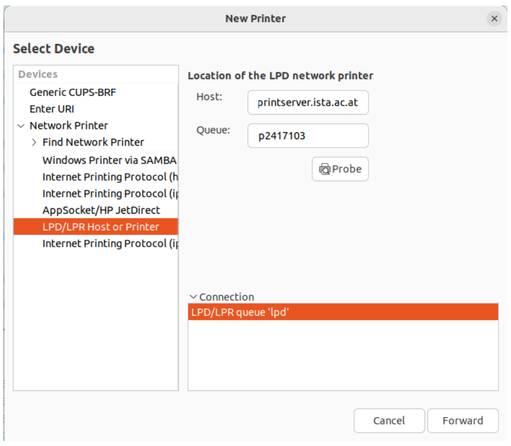
5 Installing Printer

Last edited by: Aayush Desai

For most Operating Systems and Printers please refer to this link and the links within. The full printer model name can be found atop the printer you are trying to install.

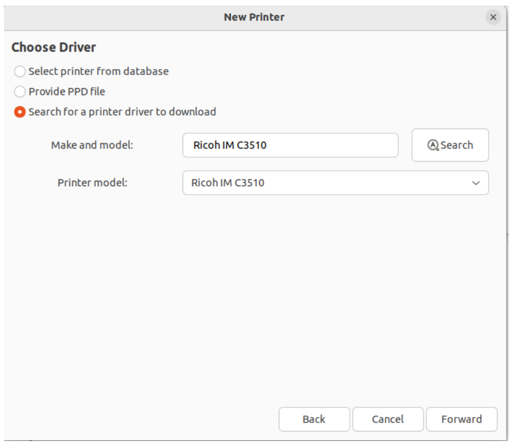
For linux users specifically; follow the steps mentioned on the website. Follow till step 2 (i.e. ”On the “New Printer” window ....”).
For step 3, try to find the printer (model name) you are looking for using the instructions on the website. If you can’t find it (Eg the ”Ricoh IM C3510”/P2417103 on 3rd floor Moonstone), then:

1. select the ”Search for a printer driver to download” option (see Fig.2b).
2. Put in the full printer model name in ”Make and Model”.
3. Once the driver is found, select it and click ”Forward”.
4. From here on, follow the instructions on the website.



Location of the LPD network printer
Host: printserver.ista.ac.at
Queue: p2417103
Connection
LPD/LPR queue 'lpd'

(a) Step 3



( ) Select printer from database
( ) Provide PPD file
[✓ Search for a printer driver to download]
Make and model: Ricoh IM C3510
Printer model: Ricoh IM C3510

(b) Step 4

Figure 2: Installing Printer