**MULTIVERSE NOTES**  (Gurpreet Matharoo)

The best way to run multiverse software is to run it directly on a cluster (Narval, for example).

The initial setup requires a Graphical user interface (GUI), which could be accessed either by:

1. Tunnelling to the cluster (see procedure to tunnel).
2. By using jupyterhub interfaces that are available on every cluster (<https://jupyterhub.narval.alliancecan.ca/>)

**Note** – My checks indicate that even though accessing the multiverse GUI is a lot easier by using jupyterhub, the resolution is however not very good. On the other hand, tunnelling, even though it is a bit of learning initially, but overall, it provides a much better resolution. The user could check both methods themselves as well.

Download the multiverse package on /home directory of the cluster. This is the default directory where the user lands up after logging on to the cluster. Then, unzip the content, and you will find **multiverse.py** file. This is a python file, and any recent version of python should be good enough to run the GUI part.

**INITIAL SETUP:**

For initial setup, these are the steps:

1. Run the command, python multiverse.py -c; this will open up the GUI. In that,
2. Seed Types = Harvard-Oxford Atlas/Saved Mask
3. Processing Mode = SLURM (if you are running this on a cluster), this will open up a further window where you setup the SLURM parameters. Then press “Set” below which takes you to the initial window.
4. You also have to “Define Seeds”, click that, and it opens a sub window, where, against “Harvard-Oxford Atlas Brain Region”, select **\*Cingulate Gyrus, posterior division\*.** Pick 50 and 90 as Minimum threshold and Maximum Threshold values, and then press, **“Add Brain Region”,** followed by pressing **“Set”.**
5. Also pick the Number of pipelines, for test purposes, set it to 1. Data generated by Graham uses 200 pipelines.
6. Finally, press **“Save Configuration”,** and that would complete the setup.

You are no setup to submit the jobs via a job submission script.

**GETTING THE APPTAINER IMAGE:**

The next step is to get the apptainer image and store it at the same location /home/username on the cluster. You can use the following command to download the apptainer image:

**apptainer pull multiverse.sif docker://gseasons/multiverse:latest**

**TO RUN THE CODE:**

python multiverse.py -r -d /Path/To/Data/Folder -o /Path/To/Output\_Folder

Modules need to be loaded: python, scipy-stack, apptainer.

Potential files to be looked at in case there are run time errors:

multiverse/configuration/batch.sh

The code needs ipyparallel to run, which needs to installed by installing a python virtual environment:

virtualenv --no-download env-pypar

source env-pypar/bin/activate

pip install - -no-index - -upgrade pip

pip install - -no-index ipyparallel

deactivate

Then, ensure that the following lines are there at the beginning on the batch.sh file,

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#!/bin/bash

module load StdEnv/2023

module load apptainer/1.3.5

module load python

virtualenv env-pypar

source env-pypar/bin/activate

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This will ensure that the virtual environment generated is setup and ipyparallel is available.

After this point, the code seems to run, however, I am still struggling with some of the paths I believe because of lack of further documentation. Assuming everything works fine, the output should be written on the output directory and data for 200 pipelines could be generated.

**PROCEDURE TO TUNNEL TO A CLUSTER NODE DIRECTLY**

As clusters are not at the local location, accessing graphical user interfaces often is challenging. While jupyterhub makes this procedure easier, there are times when you need better resolution and that is where, tunnelling is important.

You should install the latest version of TigerVNC viewer on your local desktop or laptop. Then run the following steps:

1. Open a terminal, and connect to a cluster (say, Narval). $ssh [username@narval.alliancecan.ca](mailto:username@narval.alliancecan.ca)
2. Then, run an interactive job:

salloc --ntasks=1 --time=01:00:00 --account=def-emazerol

**As a demo (I am not a part of def-emazerol), this is what I get:**

[gsingh@narval1 configuration]$ salloc --ntasks=1 --time=01:00:00 --account=cc-debug

salloc: Pending job allocation 43474743

salloc: job 43474743 queued and waiting for resources

salloc: job 43474743 has been allocated resources

salloc: Granted job allocation 43474743

salloc: Nodes nc10935 are ready for job

[gsingh@nc10935 configuration]$

**I was on login node narval1, and I get a compute node nc10935, here I will start my vncserver in the following manner:**

[gsingh@nc10935 configuration]$ module load vmd

[gsingh@nc10935 configuration]$ vncserver

Desktop 'TurboVNC: nc10935.narval.calcul.quebec:1 (gsingh)' started on display nc10935.narval.calcul.quebec:1

Starting applications specified in /cvmfs/soft.computecanada.ca/gentoo/2023/x86-64-v3/usr/bin/xstartup.turbovnc

Log file is /home/gsingh/.vnc/nc10935.narval.calcul.quebec:1.log

The vncserver has started, and I get a display :1 == port 5901

Now, open a second terminal on your laptop, and connect to the above port, use the same login node as above (narval1), to enforce connecting to narval1, we do:

ssh [gsingh@narval1.alliancecan.ca](mailto:gsingh@narval1.alliancecan.ca) -L 15901:nc10935:5901

Now open Tigervnc viewer on your desktop, and in VNC server number, type: localhost:15901, and this would open up a Narval Desktop on your local machine.

Once you are done, you should kill the vncserver that you started on the first terminal by:

vncserver -kill :1